AFRL-RI-RS-TR-2010-083
Final Technical Report
March 2010

HETEROGENEOUS VISION DATA FUSION FOR INDEPENDENTLY MOVING CAMERAS

Tennessee State University

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2010-083 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
HENRY X. SIMMONS	ROBERT S. MCHALE, Deputy Chief
Work Unit Manager	Information Systems Division
	Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* MARCH 2010	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* March 2009 – September 2009

4. TITLE AND SUBTITLE

HETEROGENEOUS VISION DATA FUSION FOR INDEPENDENTLY MOVING CAMERAS

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
FA8750-09-1-0162

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Ali Sekmen and Fenghui Yao

5d. PROJECT NUMBER
558B

5e. TASK NUMBER
TS

5f. WORK UNIT NUMBER
U2

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Tennessee State University
3500 John A. Merritt Blvd.
Nashville, TN 37209

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RISA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2010-083

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2010-1329 Date Cleared: 18-March-2010

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Image fusion problems can be classified into two categories. In *Category-I*, images obtained by sensors operating at different wavelengths and "viewing a common scene" simultaneously are fused. In *Category-II*, images collected by multiple homogenous and/or heterogeneous sensors mounted at different locations, "viewing different scenes with partial overlapping", are fused. *Category-II* image fusion is of high importance for real-time target detection, tracking, and identification over a large terrain.

The goal of the project is to investigate and evaluate the existing image fusion algorithms, develop new real-time algorithms for Category-II image fusion, and apply these algorithms in moving target detection and tracking. The research objectives are three-fold: image fusion algorithm investigation, new algorithm development, and application of the proposed algorithms to moving target detection and classification.

15. SUBJECT TERMS

Image Fusion, Target Detection, Moving Cameras, IR Camera, EO Camera

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Henry X. Simmons
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	45	**19b. TELEPHONE NUMBER (*Include area code)*** N/A

TABLE OF CONTENTS

LIST OF FIGURES

1. SUMMARY

Image fusion is important for image analysis and has important military applications such as concealed weapon detection and autonomous landing guidance. Image fusion problems can be classified into two categories. In *Category-I* image fusion, images obtained by sensors operating at different wavelengths and "viewing a common scene" simultaneously are fused. In *Category-II* image fusion, images collected by multiple homogenous and/or heterogeneous sensors mounted at different locations, "viewing different scenes with partial overlapping", are fused. *Category-II* image fusion is of high importance for real-time target detection, tracking, and identification over a large terrain. *Category-I* image fusion has been well-studied and many algorithms including image pyramid approaches, filter-shift-decimate fusion, discrete wavelet transform, and total probability density fusion have been developed. However, *Category-II* image fusion has not yet been well-studied.

The goal of the proposed project is to investigate and evaluate the existing image fusion algorithms, develop new real-time algorithms for Category-II image fusion, and apply these algorithms in moving target detection and tracking. The research objectives are three-fold: image fusion algorithm investigation, new algorithm development, and application of the proposed algorithms to moving target detection and classification.

This progress report presents several algorithms for the fusion of images in video streams collected by heterogeneous top-down cameras.

Two (2) faculty members, the Principle Investigator and the Co-Principle Investigator, and a research associate were actively involved in this research. Four (4) undergraduate students also participated for a short period of time.

2. INTRODUCTION

This work employs heterogeneous cameras for large area monitoring and target detection. To generate the view of a large area, multiple omnidirectional cameras are often used [1] [2]. The work in [1] employed four omnidirectional cameras to generate the bird-view image to monitor the surrounding area of a track-trailer. Similarly, the work in [2] used six fisheye cameras to generate the panoramic image of vehicle surroundings. The common issue for ominodirectional and fisheye cameras is that the image quality deteriorates toward the boundary of the image, and that the captured images need to be dewarped into perspective images. The work in [3] employed four pin-hole cameras to make the bird-view image of the vehicle surroundings. In this method, first, image pixels of the wide-angle cameras are back-projected to the ground plane, and then the ground points are projected to the virtual camera. The work in [4] employed a stereo camera for real-time lane and obstacle detection. Both systems need to conduct camera calibration to calculate camera intrinsic and extrinsic parameters. The problem for these methods are (i) there is no robust method for camera calibration; (ii) the small error in intrinsic camera parameter estimation brings a great distortion to the transformed images.

This work achieves the fusion of images collected by top-down heterogeneous cameras and moving target detection from the fused images. Several image fusion techniques have been reported in literature [7] [8] [9] [10]. Among these methods, discrete wavelet transform (DWT) based fusion schemes offer several advantages over similar pyramid based fusion schemes: (a) DWT provides directional information while the pyramid representation does not introduce any spatial orientation in the decomposition process [11]; (b) in pyramid based image fusion, the fused images often contain blocking effects in the regions where the input images are significantly different. No such artifacts are observed in similar DWT based fusion results [11]; and (c) images generated by DWT based image fusion have better signal-to-noise ratios (SNR) than images generated by pyramid image fusion when the same fusion rules are used [12]. Our previous work that discusses the fusion of the optical and IR images by using DWT based fusion approach is given in [5]. The difference between the current work and [5] is as follows. In work [5], both optical camera and infrared (IR) camera are mounted on a helicopter, and the camera lenses are parallel. The helicopter flies at high altitude so that the images from both cameras are bird-view images. However, in the current work, the camera lenses of optical camera and IR camera are not parallel, and the images from both cameras are top-down images but not bird-view images. This makes fusion of images from heterogeneous cameras difficult. The difference between the current work and the ones in [3] and [4] is that the later needs camera calibration to find camera intrinsic and extrinsic parameters, but our approach does not need to calculate camera parameters. The following section will describe the proposed method in detail. Section 2.3 describes another algorithm that makes use of Genetic Algorithms for finding optimum image matching parameters between IR and electro-optical (EO) images.

The report is organized as follows: Section 3 presents the algorithms developed. The experimental results and performance analysis are described in Section 4. Some conclusions are given and future work is motivated in Section 5.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Landmark-Based Algorithm Description

Figure 1 shows the camera installation and their output images. Figure 1 (a) shows the positions of four cameras, which are numbered as Camera-1, Camera-7, Camera-8, and Camera-6. Camera-1 is a dome-type optical camera (Pelco Spectra IV), Camera-7 is an IR camera (Pelco ESPRIT), Camera-8 is a hand-held camera (Sony Handycam DCR-PC10), and Camera-6 is a mailbox-type optical camera (Pelco ccc1390H-6). The distance between Cameras 1 and 7 is 19.2m, Cameras 7 and 8 is 13.4, Cameras 8 and 6 is 13.4 m, and the altitude from the ground is 8.6m. Figure 1 (b), (c), and (d) show the output image of Camera 6, 7, and 8, respectively. The output images from Camera 1, 6, and 8 are the same, 640×480 color image, and the output image from camera 7 is 320×240 dummy-color image (R-, G-, and B-channel are the same luminance component). Although Camera 1, 6 and 8 are all optical cameras, their optical characteristics are different. Besides these differences, the images from Camera 1, 6, 7 are recorded in YUYV format, and the image from Camera 8 is recorded in RGB format. Therefore, the output images are different in color and luminance. The following explains the major components of the proposed algorithm: image rectification, image interpolation, image fusion, and moving target detection.

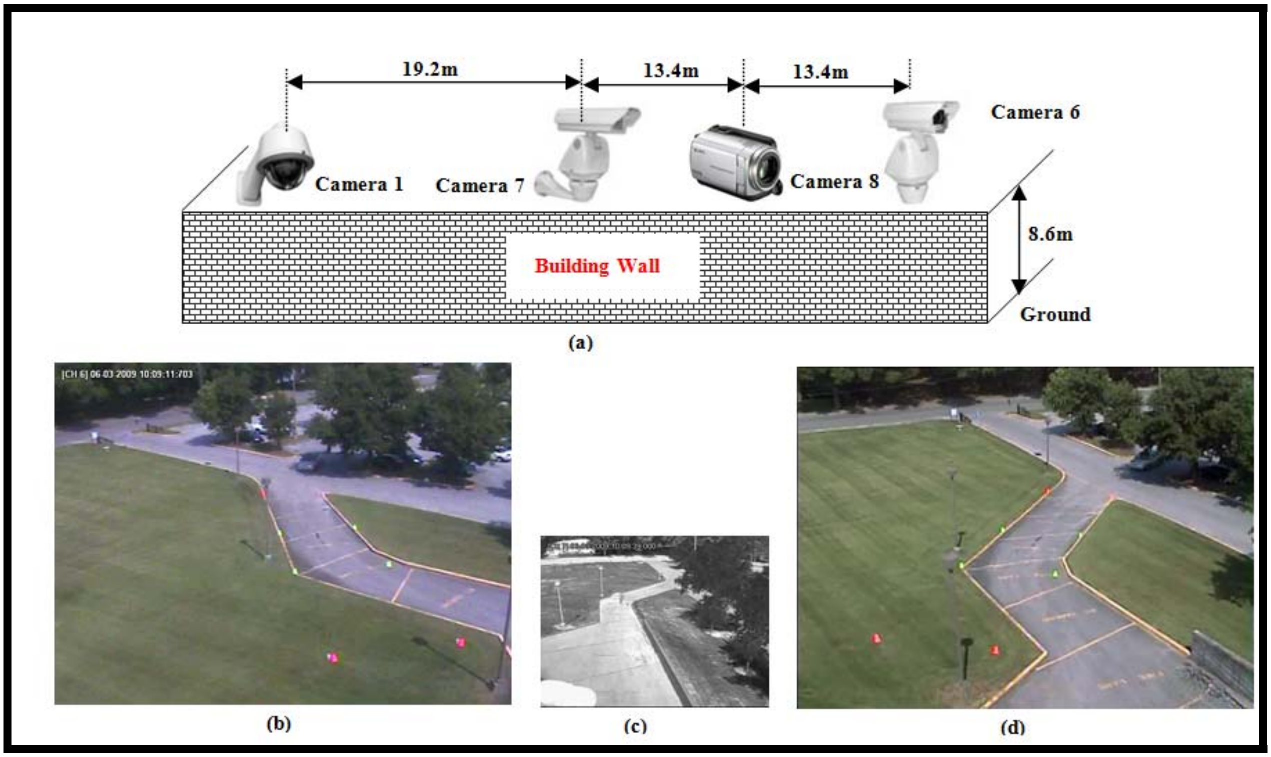


Figure 1: (a) Camera 1, 7, 8, and 6, Installed at the Top of the Building Wall; (b) Optical Image from Camera-6; (c) IR Image from Camera-7; (d) Optical Image from Camera-8

3.1.1 Image Rectification

As shown in Figure 1, the images from the heterogonous cameras are distorted because the cameras are top-down cameras. To fuse the images from heterogeneous top-down cameras, the images from these cameras needs to be registered. There are many methods for image registration such as region-based image registration [16], contour-based image registration [17], and implicit similarity based registration [18]. However, these methods are applicable for aerial image registration or microscopy medical image registration, in which camera lenses are parallel. They are not applicable for registering top-down images, where camera lenses are not parallel, and regions and contours are greatly distorted. In this work, we introduce rectification-based method for registration. Image rectification is a transformation process used to project multiple images onto a common image surface. It is used to correct a distorted image into a standard coordinate system. It can be done by calculating camera intrinsic and extrinsic parameters. However this method is computationally heavy, and is not robust to camera parameter estimation error. In our work, we employ the projective transformation [6]. Let $\mathbf{m'} = (x', y', w')^T$ denote the distorted pixel position, and m = $(x, y, 1)^T$ the distortion free pixel position, then the relation between m′ and m is shown by $\mathbf{m' = Hm}$, that is,

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

where w' is the scale factor. The projective transformation matrix **H** (or homography matrix) has 8 degree of freedom. As shown in Figure 2, assume the vertex coordinates of a square *ABCD* is $A(x_1,y_1)$, $B(x_2,y_2)$, $C(x_3,y_3)$, and $D(x_4,y_4)$, respectively, their transformed coordinates are $A'(x'_1,y'_1)$, $B'(x'_2,y'_2)$, $C'(x'_3,y'_3)$, and $D'(x'_4,y'_4)$, correspondingly, the 8 parameters of the homography matrix **H**, from the square *ABCD* to the quadrilateral *A'B'C'D'*, is determined by solving the following equations,

$$\begin{pmatrix} x'_1 & y'_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -y'_1y_1 \\ 0 & 0 & 0 & x'_1 & y'_1 & 1 & -x'_1y_1 & -y'_1y_1 \\ x'_2 & y'_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -y'_2y_2 \\ 0 & 0 & 0 & x'_2 & y'_2 & 1 & -x'_2y_2 & -y'_2y_2 \\ x'_3 & y'_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -y'_3y_3 \\ 0 & 0 & 0 & x'_3 & y'_3 & 1 & -x'_3y_3 & -y'_3y_3 \\ x'_4 & y'_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -y'_4y_4 \\ 0 & 0 & 0 & x'_4 & y'_4 & 1 & -x'_4y_4 & -y'_4y_4 \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{pmatrix} \tag{2}$$

where $a_{33} = 1$.

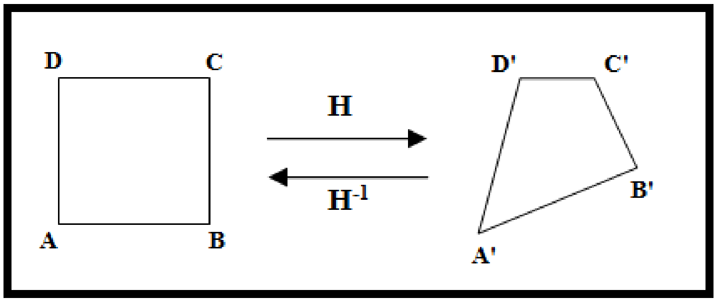


Figure 2: Square ABCD is Transformed to a Quadrilateral A'B'C'D', and Its Inverse Transformation

To solve Eq. (2), it needs four image point pairs from the corresponding images. Finding this quadruplet is a typical and hard problem in computer vision. There are many methods to find image point pairs from two images. But none of them is reliable. Especially in this work, the cameras are heterogeneous, their optical characteristics are different, and image formats are different (that is, image qualities are different). It is very difficult to find this quadruplet automatically. This work employs the edges and edge corners as the landmarks. The quadruplet is determined by selecting the corresponding landmarks manually. This approach is applicable because it is needed only once at the beginning of the processing. The landmarks in a video stream can be tracked automatically in the same video stream. Figures 3 (a), (b), and (c) show the input images from camera 6, 8, and 7, respectively. The landmark candidates are defined as the intersection of two lines, and are marked by blue dots in Figures 3 (d), (e), and (f), where the lines are detected by Hough transform. The landmark physical positions and the landmark perimeter size can be obtained from Google earth map. Figure 4 shows the 6 landmarks numbered from 1 to 6 (the red lines are auxiliary to help understanding the definition of landmarks). Distance between landmark 1 and 2, landmark 1 and 5 is 19.11m and 29.25m, respectively, the width of this region is 5.0m, which are measured from Google earth map. The quadruplet selection from these candidates in Figures 3 (d), (e), and (f), can be {0, 2, 3, 1}, {2, 12, 9, 0} and {25, 29, 23, 19}, or {4, 2, 3, 5}, {6, 12, 9, 4}, and {14, 29, 23, 13}, or {0, 4, 5, 1}, {2, 6, 4, 0}, and {14, 13, 19, 25}, from camera 6, 8, and 7, respectively. The area circled by the landmarks are called landmark region. Figures 5 (a), (b), and (c) show the rectification results for image in Figures 3 (a), (b), and (c), respectively.

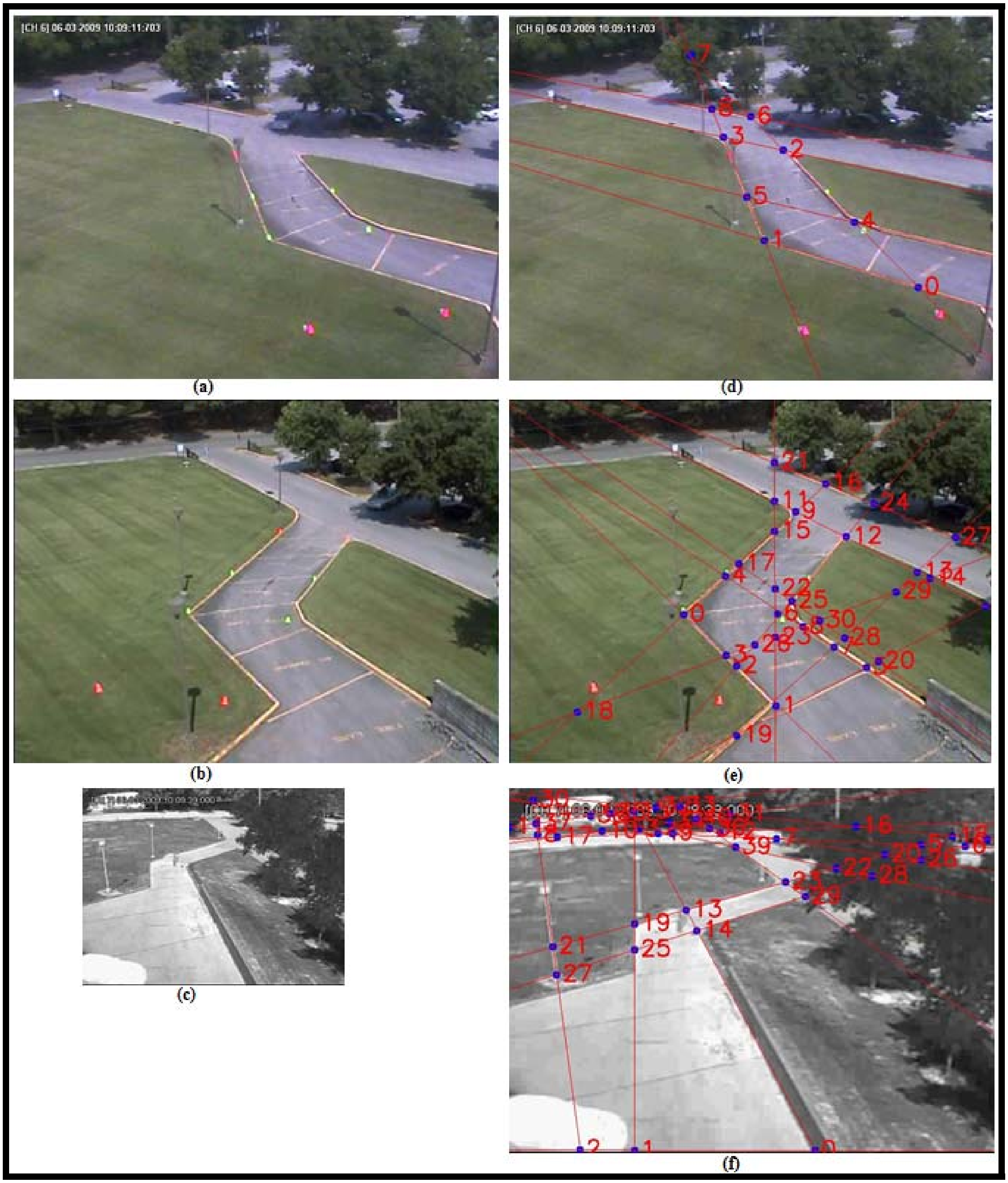


Figure 3: (a), (b), and (c) Images from the Optical Camera 6, 8, and IR Camera 7. The Size of Optical Images is 640×480, and IR Image is 320×240. (d), (e), and (f) Landmark Candidates Detected from the Corresponding Input Image

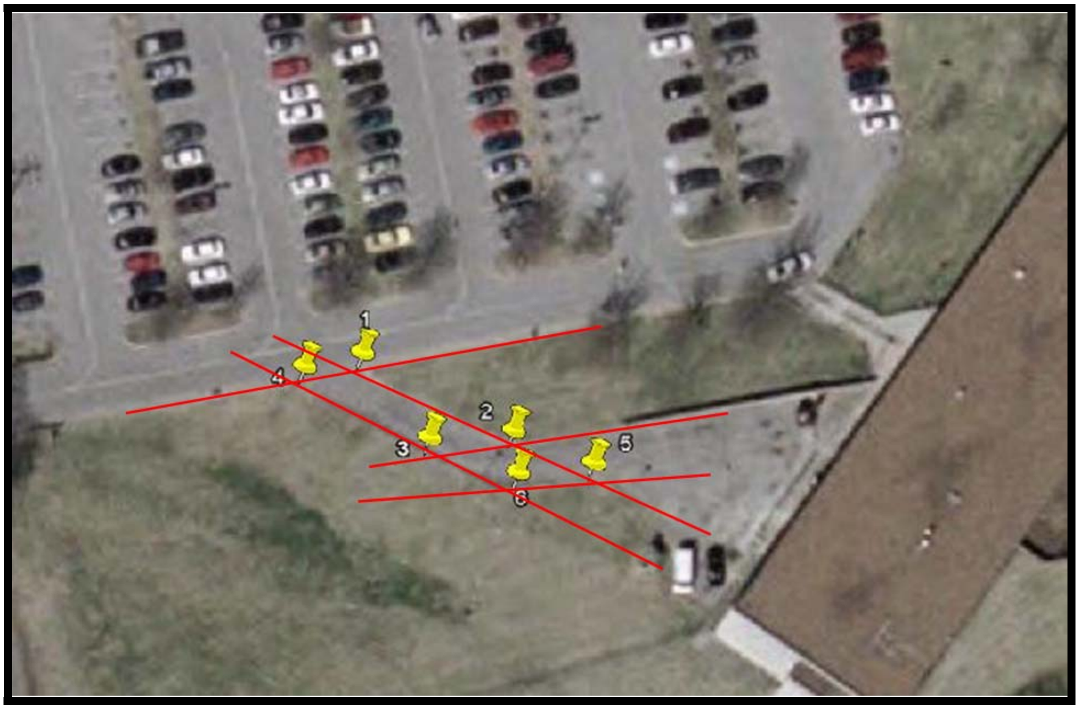


Figure 4: Physical Positions of Landmarks and Length and Width of the Landmark Region from Google Earth Map. Distances between Landmark 1 and 2, Landmark 1 and 5 are 19.11m and 29.25m, Respectively, the Width of this Region is 5.0m, which are Measured from Google Earth Map

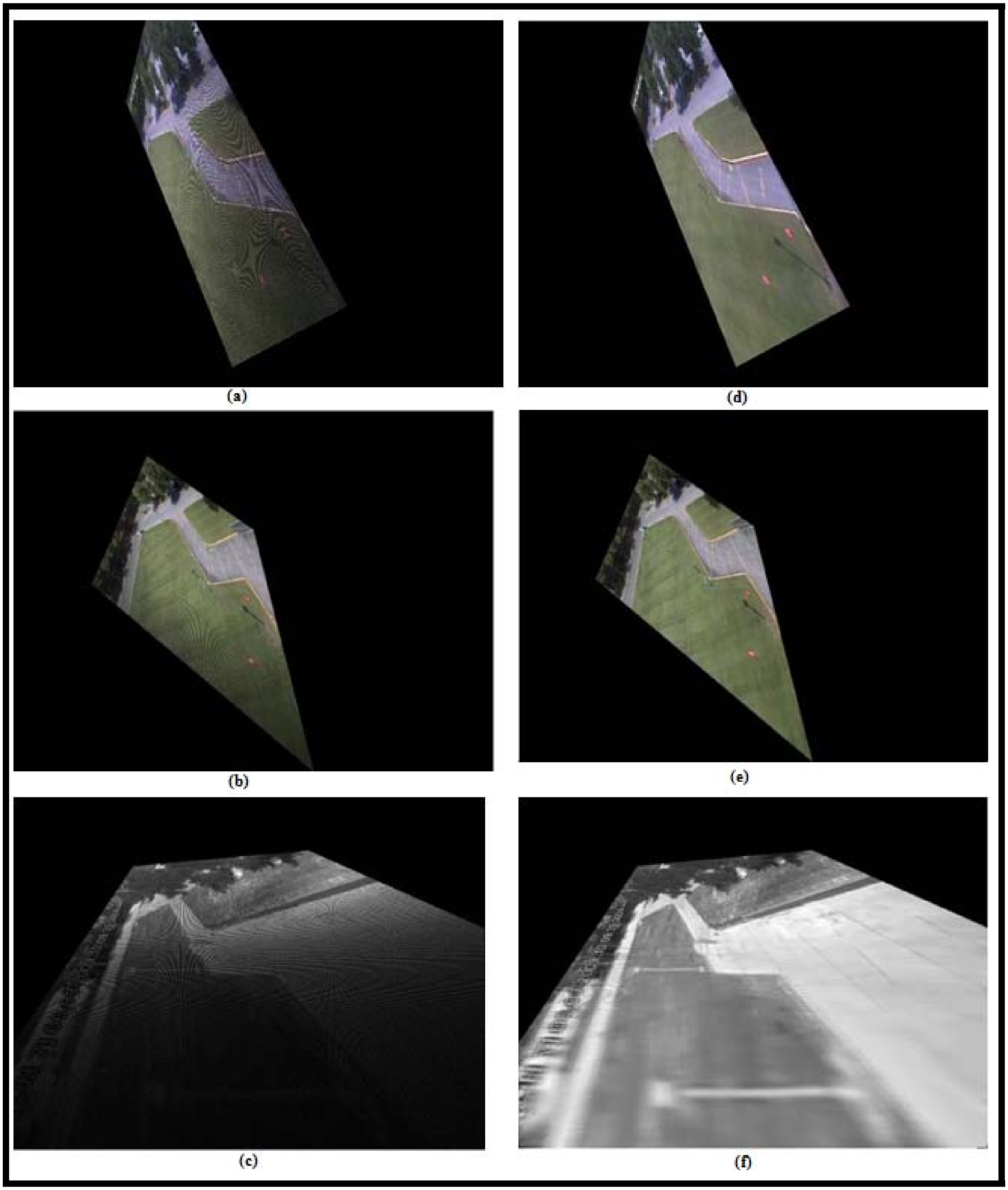


Figure 5: (a), (b), and (c) Rectification Results (no Interpolation) for Input Image in Figure 3 (a), (b), and (c), Respectively. (d), (e), and (f) Interpolation Results

3.1.2 Image Interpolation

When an image in transformed by a transformation model, there will always be a difference between the input and transformed images. This difference is called the residual (or error). The residuals in Figures 3 (d), (e), and (f) are the black dots in the image region. The image data at residual points can be generated by employing image interpolation techniques. There are many image interpolation methods [19]. The method used in this work is simple and robust, which is described as follows.

(i) For an image line, search from left to right. Mark the first point where image data is non-zero as m_1.
(ii) Next, search the point where the image data first becomes zero, and mark the point right before this point as m_2.
(iii) Then, search the point where the image data first becomes non-zero again, and mark this point as m_3. For all points between m_2 and m_3, set their image value by using the average of image value at m_2 and m_3.
(iv) Repeat above for all image lines.
(v) Repeat (i) to (iv) for R-, G-, and B-component for whole image.

The image interpolation results for the rectified image in Figures 5 (a), (b), and (c) are shown in (d), (e), and (f), respectively.

3.1.3 Image Fusion

The successful fusion of images acquired from different modalities or instruments is of great importance in many applications, such as medical imaging, microscopic imaging, remote sensing, computer vision, and robotics. Image fusion can be defined as the process by which several images or some of their features are combined together to form a single image. Image fusion can be performed at four different levels of the information representation, which are signal, pixel, feature, and symbolic levels. Several image fusion techniques have been reported in literature [7] [8] [9] [10] [11] [12]. As related in Section 2, DWT-based methods have some advantages over other methods; our fusion algorithm employs DWT based method. In DWT-based image fusion schemes the wavelet transforms W of the two registered input images $I_1(m,n)$ and $I_2(m,n)$ are computed and these transforms are combined using some kind of fusion rule φ. Then, the inverse wavelet transform W^{-1} is computed and the fused image $I(m,n)$ is reconstructed, that is,

$$I(m,n) = \mathbf{W}^{-1}(\varphi\ (\mathbf{W}(I_1(m,n)), \mathbf{W}(I_2(m,n)))) \tag{3}$$

Among Harr wavelets, Gabor wavelets, and Daubechies wavelets, our approach employs Daubechies wavelets [13] [14]. Let us consider the input image $f(m,n)$ as the scaling coefficient, $s^{0}_{m,n}$, at scale 0, then 2-D DWT at scale j is denoted by,

$$s^{(j+1)}_{m,n} = \sum_{l}\sum_{k} \overline{p_{k-2m}}\ \overline{p_{l-2n}}\ s^{(j)}_{k,l}\ , \tag{4.1}$$

$$w^{(j+1,h)}_{m,n} = \sum_{l}\sum_{k} \overline{p_{k-2m}}\ \overline{q_{l-2n}}\ s^{(j)}_{k,l}\ , \tag{4.2}$$

$$w_{m,n}^{(j+1,v)} = \sum_{l} \sum_{k} \overline{q_{k-2m}}\ \overline{p_{l-2n}}\ s_{k,l}^{(j)}\ , \tag{4.3}$$

$$w_{m,n}^{(j+1)} = \sum_{l} \sum_{k} \overline{q_{k-2m}}\ \overline{q_{l-2n}}\ s_{k,l}^{(j)}\ . \tag{4.4}$$

And the reconstruction is given by

$$s_{m,n}^{(j)} = \sum_{l} \sum_{k} \big(p_{m-2k}\, p_{n-2l}\, s_{k,l}^{(j+1)} + p_{m-2k}\, q_{n-2l}\, w_{k,l}^{(j+1,h)} + q_{m-2k}\, p_{n-2l}\, w_{k,l}^{(j+1,v)} + q_{m-2k}\, q_{n-2l}\, w_{k,l}^{(j+1,d)} \tag{5}$$

where p_k is Daubechies series, and $q_k = (-1)^k p_{-k}$. Applying Eq. (4) to image *f*(*m*,*n*) produces four subbands, $s_{m,n}^{(j+1)}$, $w_{m,n}^{(j+1,h)}$, $w_{m,n}^{(j+1,v)}$, and $s_{m,n}^{(j+1,d)}$. By recursively applying the same scheme to $s_{m,n}^{(j+1)}$, a multiresolution decomposition can be achieved. At each scale, the subbands are sensitive to frequencies at that scale, and $w_{m,n}^{(j+1,h)}$, $w_{m,n}^{(j+1,v)}$, and $s_{m,n}^{(j+1,d)}$ are sensitive to horizontal, vertical, and diagonal frequencies, respectively.

Let B_i={ $s_{m,n}^{(j+1)}$, $w_{m,n}^{(j+1,h)}$, $w_{m,n}^{(j+1,v)}$, $s_{m,n}^{(j+1,d)}$} denote DWT subbands of *i*-th image, the fusion rule φ in Eq. (3) can be rewritten as

$$\kappa_1 I_{1,wavelet} + \kappa_2 I_{2,wavelet} \tag{6}$$

where $I_{i,wavelet} \in B_i$ (i = 1, 2), and κ_1 and κ_2 are weighting coefficients. By tuning κ_1 and κ_2, DWT based fusion can emphasize or depress some directional components in input images I_1(*m*,*n*) or I_2(*m*,*n*). In comparison, the fusion that directly employs the input image I_1(*m*,*n*) and I_2(*m*,*n*) is given by linear conjugation as,

$$I(m,n) = \kappa_1 I_1(m,n) + \kappa_2 I_2(m,n) \tag{7}$$

in which there is no way to emphasize or depress some directional components included in I_1(*m*,*n*) or I_2(*m*,*n*). Figure 6 shows image fusion result for the interpolated image in Figures 5 (d), (e), and (f).

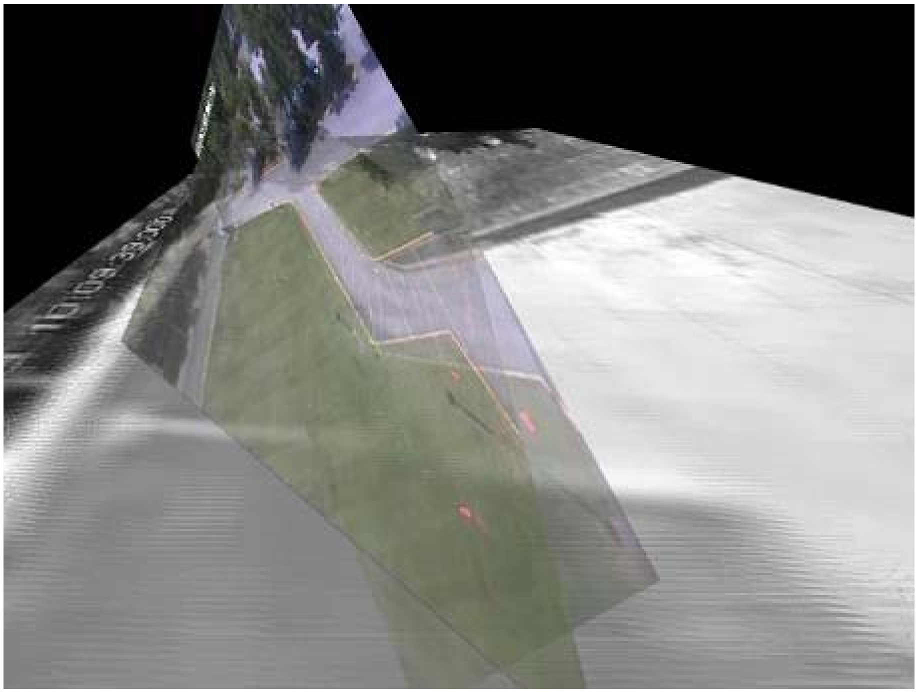


Figure 6: Image Fusion Result for the Interpolated Image in Figure 5 (d), (e), and (f)

3.1.4 Moving Target Detection

The targets in our system are the moving vehicles or pedestrians. The frame differencing and adaptive background subtraction are popular techniques for target detection. Because the cameras in our system are mounted on the top of the building wall, and to monitor large area, they are usually not zoomed up. This means that the movement of tree leaves and other small objects are relative small in comparison with the vehicles or pedestrians. Under this assumption, the frame differencing is fast and effective even in outdoor environment. However, the signal-to-noise ratio in frame difference is low, and it is difficult to extract the target directly from the frame difference. Here, an enhancement processing is needed. Our previous work in [15] employed the dynamic Gabor filter to enhance the frame difference images for the video streams generated from the moving platform, in which the optical flows are used to determine the orientation of the Gabor filter. In this work we introduce the *integrated Gabor filter* to generate the mask image for target extraction, because the cameras are considered as the stationary camera in the current system.

The integrated Gabor filter is defined as follows. A Gabor wavelet is defined as,

$$\psi_{\mu,\nu}(z) = \frac{\left\|k_{\mu,\nu}\right\|^2}{\sigma^2} e^{-\frac{\left\|k_{\mu,\nu}\right\|^2 \times \left\|z\right\|^2}{2\sigma^2}} \left[e^{ik_{\mu,\nu}z} - e^{-\frac{\sigma^2}{2}} \right] \qquad (8)$$

where $z = (x, y)$ is the point with the horizontal coordinate x and the vertical coordinate y. The parameters μ and ν define the orientation and scale of the Gabor kernel, $\|\cdot\|$ denotes the norm operator, and σ is related to the standard derivation of the Gaussian window in the kernel and determines the ratio of the Gaussian window width to the wavelength. The wave vector $k_{\mu,\nu}$ is defined as follows

$$k_{\mu,\nu} = k_\nu e^{i\phi_\mu} \tag{9}$$

where $k_\nu = k_{\max} / f^\nu$ and $\phi_\mu = \pi\mu/8$, $k_{\max}$ is the maximum frequency, and f^ν is the spatial frequency between kernels in frequency domain. The integrated Gabor filter is given by,

$$I_{mask} = \int_\mu \psi_{\mu,\nu}(z) * I(m,n) \tag{10}$$

where μ is the orientation of the Gabor filter, $k_{\max}$, σ, f, and ν are fixed at $\pi/2$, 2π, $\sqrt{2}$, and 3, respectively. The target detection results are shown in the following section.

3.2 Matching-Based Algorithm Description

This work shows the fusion of images generated by an optical camera and an IR camera mounted on a building, and the target detection from the fused images. Figure 7 shows the samples of an optical image and an IR image.

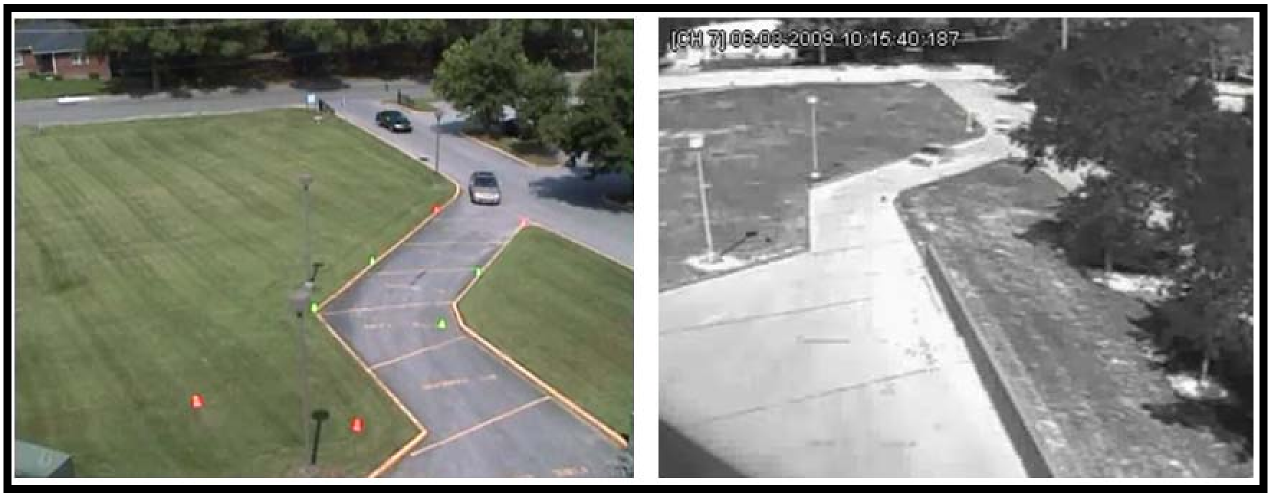


Figure 7: Two Different Images. (a) 640×480 Optical Image; (b) 320×240 IR Image

To clarify the fusion problem, we can summarize the facts and the assumptions related to the algorithm as:

- Everything including background in the optical camera is moving since we moved the camera when gathering the data.
- The optical image is color image, and IR image is grayscale image but recorded as pseudo color image, i.e., IR signature is recorded to R-, G-, and B-channels. The resolution is different (640×480 for optical image and 320×240 for IR image), and the ratio of width to height is different;
- There are some region overlaps; however, those regions are unknown;
- There are multiple targets in images, and the number of targets may change (exit or reenter the field of view of a camera).
- The cameras are very close to ground so it is impossible to use simple 2D based algorithms like Sheikh et.al used.
- In 3D environment the occlusions play more role than in 2D environments.
- The cameras, we used were uncelebrated, so we searched for robust algorithms to struggle with this problem also.

To address these problems, we designed a silhouette-based image fusion and target detection algorithm. The entire processing flow is shown in Figure 8. This algorithm consists of image transform, image registration, image fusion, and target detection. This research doesn't assume that multiple cameras are mounted on the same helicopter or an unmanned aerial vehicle (UAV) or the same building. Therefore, it is only necessary to perform the image registration every time using certain number of image sequences to determine a relative motion relation between the optical and IR cameras. The following explains these three components in detail.

Simply the algorithm takes one optical image taken by a moving optical camera and one IR image taken by a static camera. The motion template algorithm needs static images to perform, so we compensated the motion in the optical image by using the feature point extraction, optical flow detection, global parametric motion model estimation, motion detection and compensating this motion. This motion template algorithm works very robust and creates silhouettes for the moving objects both in IR and optical images. Than the silhouettes of the IR image is exposed to many perspective transformation, the characteristics of which are optimized by Genetic Algorithms (GA). All these transformed images are fused by optical image to find the best matching one. After the matching, the real images fused and the objects are tracked.

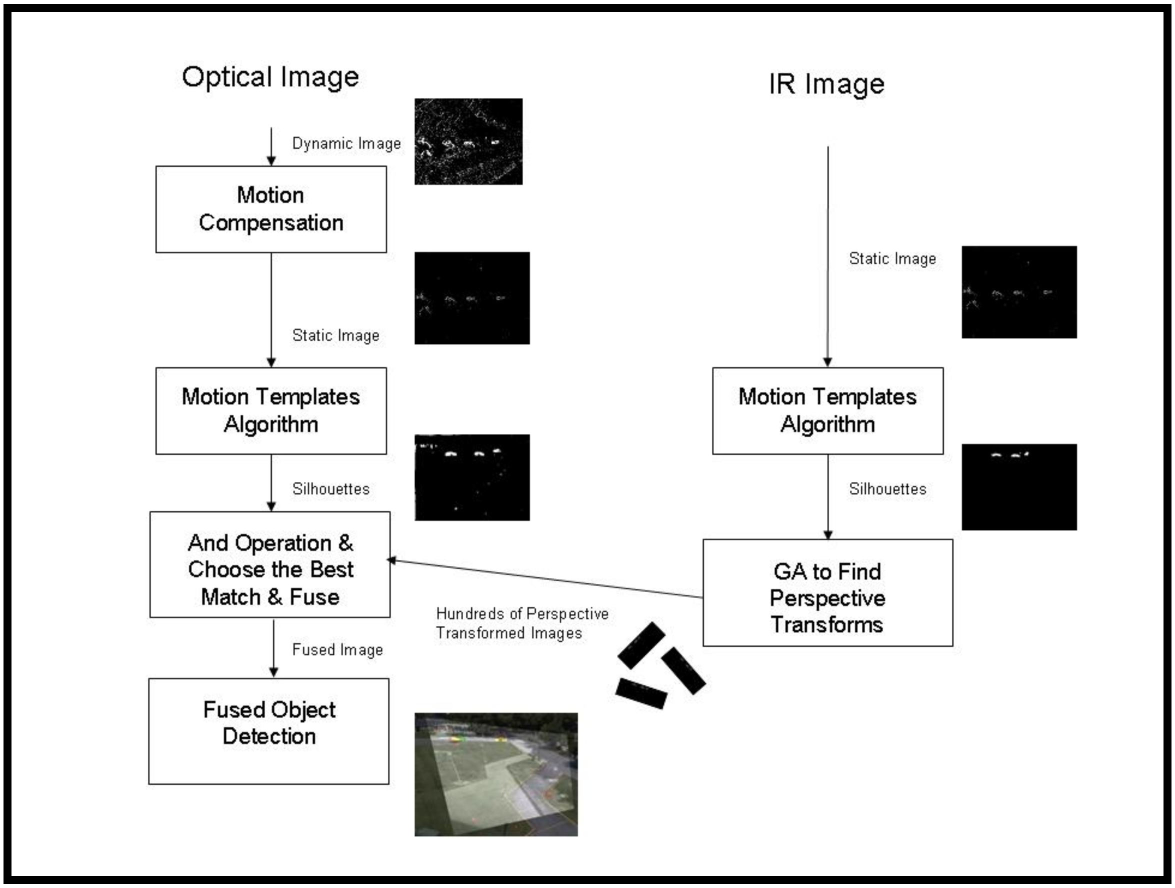

Optical Image
Dynamic Image
Motion Compensation
Static Image
Motion Templates Algorithm
Silhouettes
And Operation & Choose the Best Match & Fuse
Fused Image
Fused Object Detection
IR Image
Static Image
Motion Templates Algorithm
Silhouettes
GA to Find Perspective Transforms
Hundreds of Perspective Transformed Images

Figure 8: Process Flow of the Entire Algorithm

3.2.1 Motion Estimation and Compensation

In order to work with the motion templates algorithm, we need to compensate the motion. The motion compensation contains the feature point extraction, optical flow detection, global parametric motion model estimation, and motion compensation.

We have used Shi and Tomasi’s algorithm for feature point extraction. We have used Lucas-Kanade optical flow in a pyramid for moving feature point detection and finally affine transform the image to compensate the motion.

3.2.2 Motion Templates Algorithm

With the introduction of inexpensive/powerful hardware and increasing interest in wireless interfaces, interactive environments, tracking/surveillance systems and entertainment domains, computer vision has focused on understanding and recognizing action more; and various approaches, that attempt the full three-dimensional recognition of motion, have grown at an increasing rate.

Bobick and Davis [20, 21] in MIT Media Lab invented an effective way of tracking motion. They started with a solution to recognize action in extremely blurred images, which includes the idea of recognizing action from the motion itself, as opposed to constructing three-dimensional images first [21]. Specifically their recognition theory is based on first describing where the motion is in the spatial pattern and then describing how it is moving. Furthermore, they developed this approach into a novel representation and recognition technique for identifying actions which is based on *Temporal Templates* and their dynamic matching in time [22]. Specifically their approach is based on:

The binary representation of the localization of the motion has occurred (a motion-energy image (MEI) and motion-history images (MHI) which are a scalar-valued image, the intensity of which is a function of motion history. In our algorithm we used image silhouetted to represent patterns of moving objects. When a new frame comes, the existing silhouettes are decreased in value subject to a threshold value of 30 and the new silhouette is overlaid at maximal value. This layered motion image has the advantage that a range of times from frame to frame to several seconds may be encoded in a single image. One of the novelties of our research appears here. We are not only taking the path of the moving objects [23] or assuming the appearances of the model is invariant to 2-D rigid transformation and scaling [24] but also the appearance of the moving object as silhouettes. In Figure 9, the way to create the silhouettes is shown. As seen in Figure 10, working in the silhouette-domain helps us to find the relationship between the images. In the next section, the way to find this relationship will be discussed.

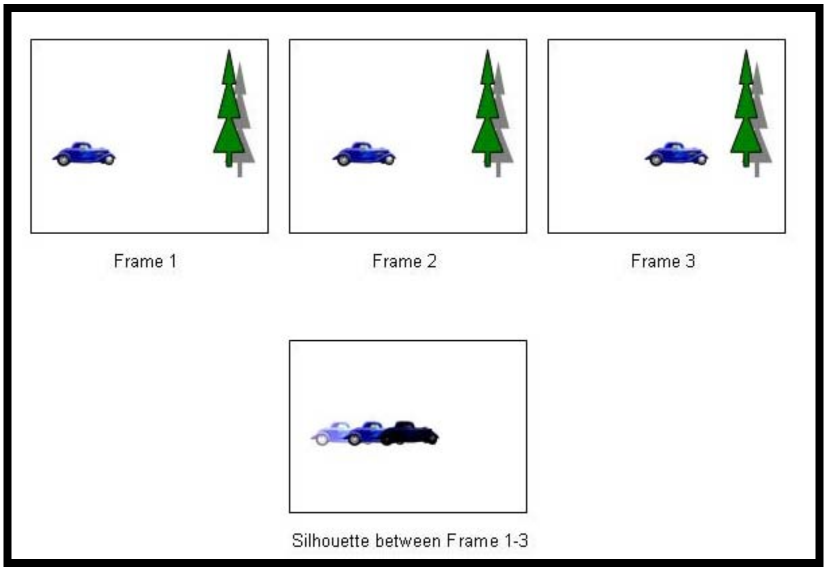


Figure 9: Forming Silhouettes

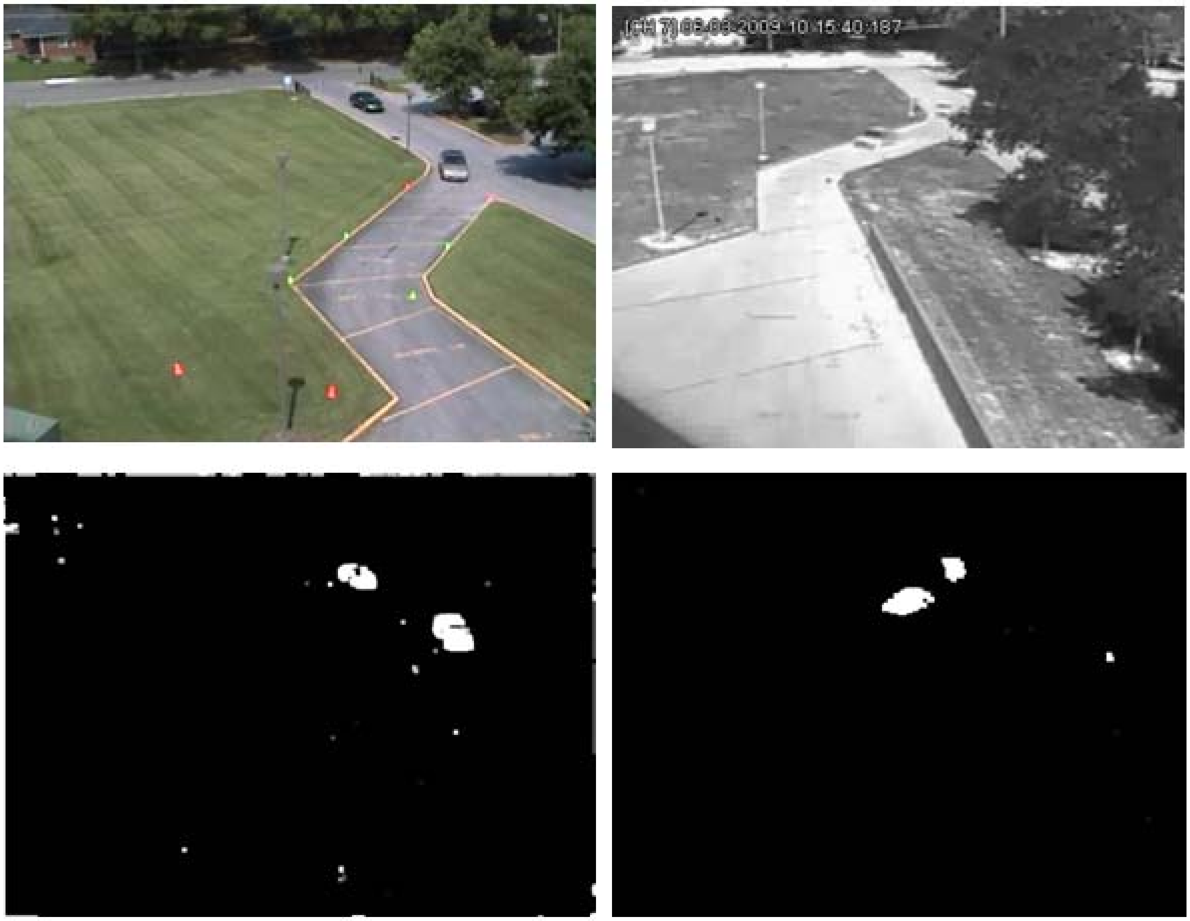


Figure 10: Forming Silhouettes from IR and Optical Images

3.2.3 Genetic Algorithm and Perspective Transformations

Perspective transformation is a method for computing the way in which a plane in three dimensions is perceived by a particular observer, who might not be looking straight on at that plane or simply perspective transformation produces perspective by viewing the 3-D space from an arbitrary eye point. In this section, we will explain how we perspective transform the IR silhouette so that it overlaps the optical image silhouette. There are two ways to find this transformation matrix.

1) Find the parameters in the transformation matrix
2) Find arrays of four points so we can independently control how the corners of a rectangle in IR silhouette data are mapped to optical silhouette.

We have tried both of them. We couldn't get control on the first one, because it is impossible to put constraints on a transformation matrix, so we couldn't find a good transform matrix out of infinite number of matrices. On the other hand, the second choices enable us to limit the search space by simply introducing some constraints. The idea behind this transformation is show in Figure 11.

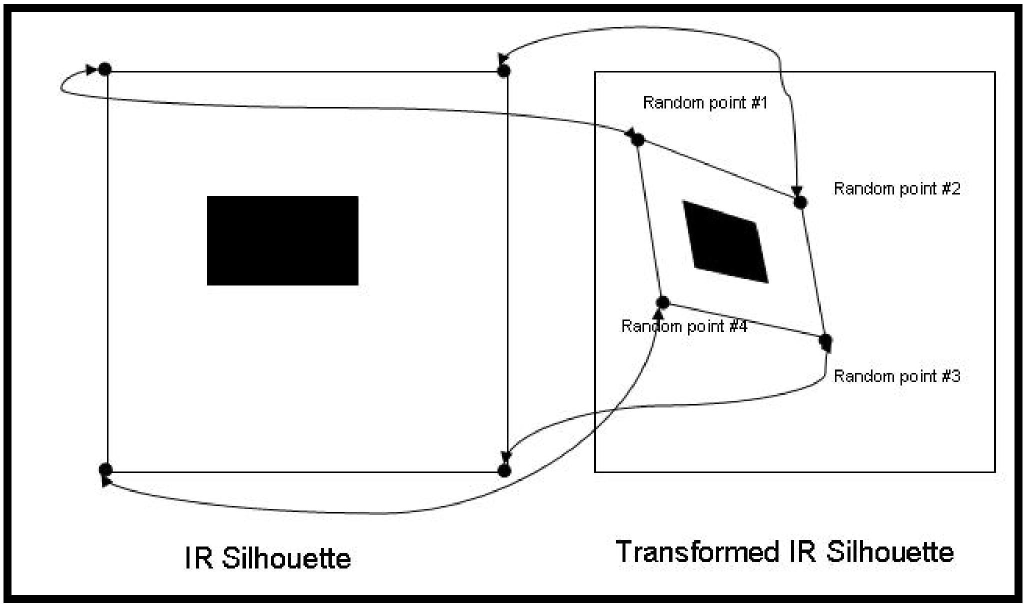


Figure 11: Selecting Points to Transform the IR Silhouette Data

Randomly selected four points correspond to the corners of the IR silhouette image and all the transformation is done accordingly. This time there is finite number of choices (width*height)^4. For example, there may be a case as shown in Figure 12.

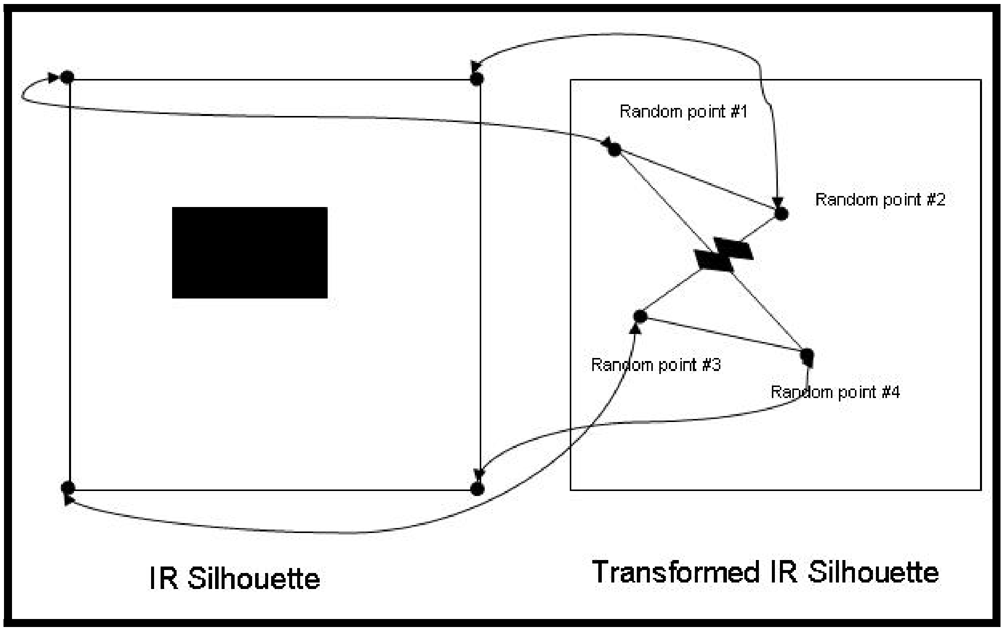


Figure 12: Undesired Points to Transform the IR Silhouette Data

One of the ways to put the points in order is using Graham Scan Algorithm. The Graham Scan, proposed by Ronald Graham (1972), is an efficient algorithm for planar convex hull. It basically computes the convex null of a given set of points on the plane with time complexity *O(n log n).*

The algorithm works in three phases: First we find the point with the lowest y-coordinate, which is also called P, the pivot. Next, we sort the points in the order of increasing angle about the pivot with time complexity O(n log n), which is a general-purpose sorting algorithm. An additional point to draw attention is that it is not necessary to calculate actual angles the points make with x-axis and it is sufficient to calculate the cotangent of the angles. Next, for each of the points, we need to determine whether moving from two previous conditions to the present condition is a left turn or a right turn; which means that the interior points on the ray cannot be part of the convex hull and should be removed. This process continues as long as the last three points are right turn and the algorithm moves on to the next sorted point as soon as a left turn is necessary. In other words, we build the null by moving around the convex, adding edges when we make a left turn and back-tracking when we make a right turn. Finally, the process ends up with the starting point with a stack that contains the points on the convex null in counterclockwise order. By this way, we are sure that a case like the one shown in Figure 12 will not happen.

This algorithm works well but slows down the process. We have found another constraint which can be applied to our data but does not work as general as Graham Scan. We randomly select points from each quadrant as shown in Figure 13.

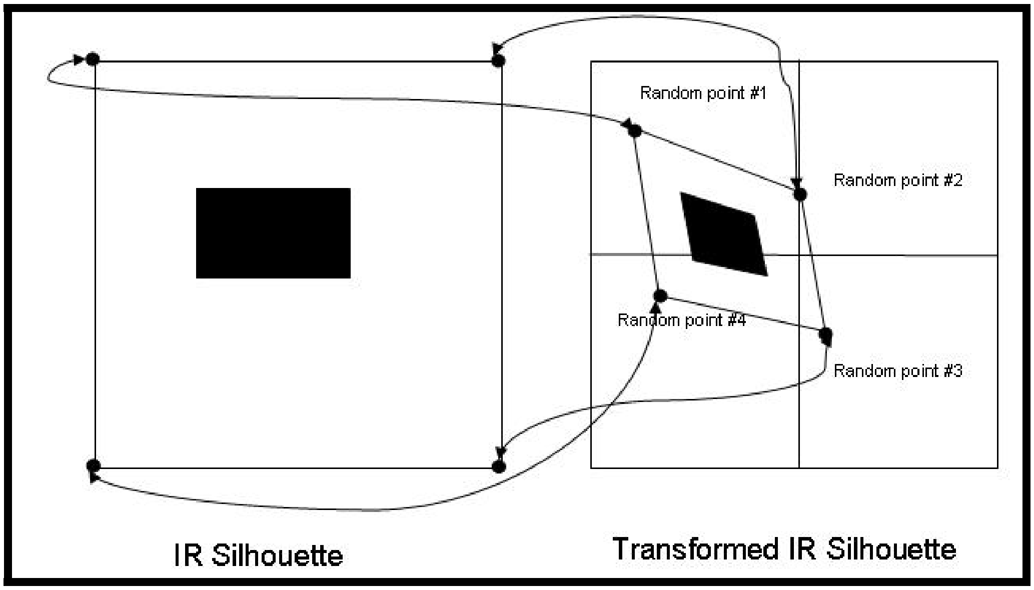


Figure 13: Only One Point from Each Quadrant is Selected to Transform the IR Silhouette Data

This is sufficient for the convexity of the points but necessary to be a good candidate for the transformation. So we used genetic algorithms to find one of the best solutions very fast, because it can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear.

The objective or the performance criteria is to maximize the number of overlapping pixel between the transformed IR silhouette image and optical silhouette image. One of the important issues is that; sometimes only counting the overlapping pixels is not enough, because the transformation can put IR silhouette image into an anomaly magnified form. So the fitness function or the objective criteria is set to number of overlapping pixel / number of nonzero pixels in the transformed IR silhouette image. In the population stage, 100 members (4 point arrays) are produced by assigning four points randomly to them. In the mate stage, the best members randomly interchange the points and create new members. Iteration number is 8. in the mutation stage, one of the points of the members changes randomly. The process is shown in Figure 14.

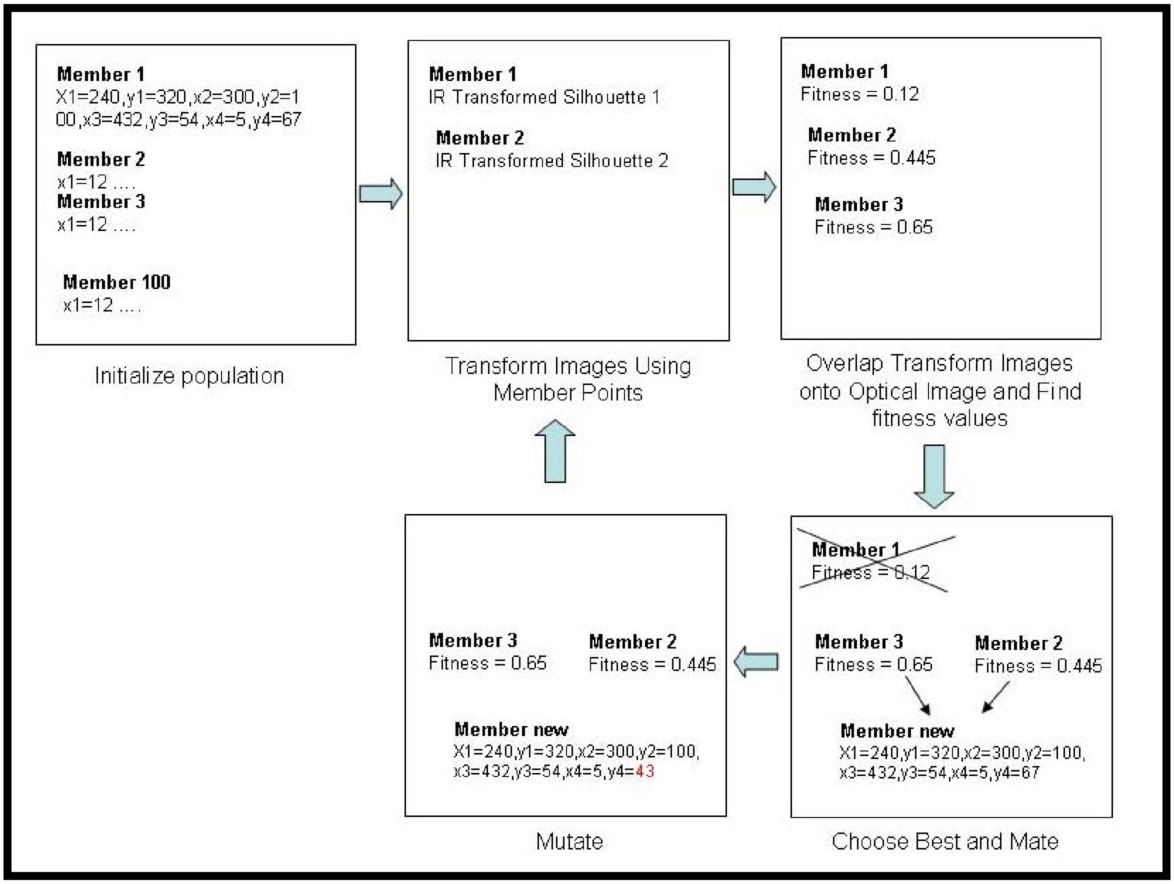


Figure 14: GA Optimization to Find the Best Overlapping Transform of IR Silhouette Data to the Optical Silhouette Data

3.2.4 Fusion and Object Detection

The member which has the highest fitness value is taken out of the population and its transformation is applied to the IR silhouette image. The resulting image is then overlapped onto the optical image. The overlapping points are the points that belong to the same moving object. The procedure is shown in Figure 15.

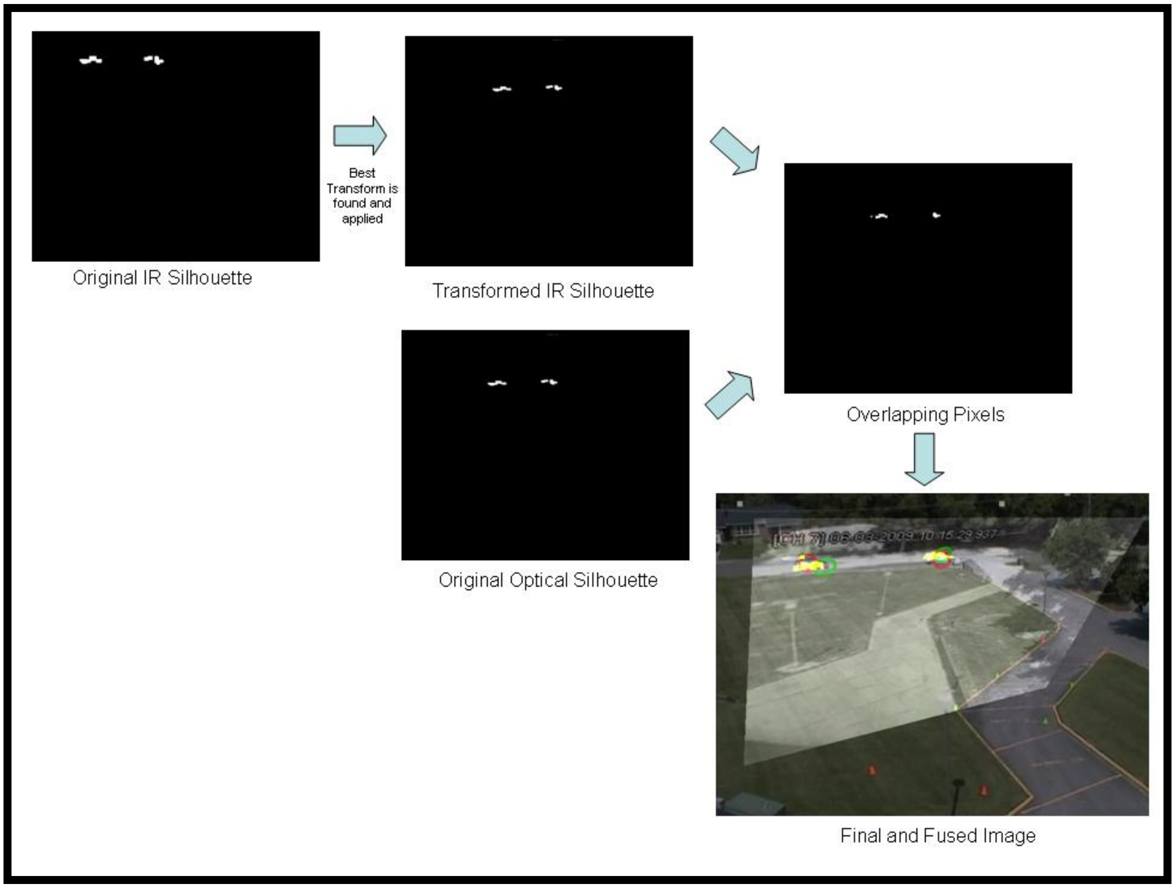

Original IR Silhouette
Best Transform is found and applied
Transformed IR Silhouette
Overlapping Pixels
Original Optical Silhouette
Final and Fused Image

Figure 15: Transformation and Fusing Process

The red circles are showing the objects detected from the optical image silhouette and the green circles are showing the objects detected from the IR image silhouette. The yellow pixels are the overlapping pixels between the optical silhouette and transformed IR silhouette.

3.2.5 Graphic processing Unit (GPU) Implementation of the Algorithm

Real-time data acquisition and processing, which are highly concurrent and distributed, require a considerable amount of computing time and power. This computing time can be reduced by distributing a process over several processors, or the computing power can be distributed among the tasks according to their priorities by taking the advantage of multi-processor hardware architectures such as GPU. GPUs are dedicated graphics rendering devices for computers, cheap enough to be affordable by everyone, powerful enough to be able to replace clusters of tens of modern computers, very efficient at manipulating and displaying computer graphics. Their highly parallel hardware architecture makes them more effective than general-purpose central processing units (CPUs) for a range of complex algorithms such as image or video processes.

Processes like real-time computer vision algorithms for detection, identification, and tracking of moving targets in video streams can fully benefit from GPU-based hardware and programming models, thus taking advantage of additional features such as the availability of parallel, fast and accurate functions or instruction sets, as well as the compact size and relatively low cost of these units.

Although much progress has been made in the parallel programming over the last several decades, the study of implementing these parallel algorithms to real-time computer vision algorithms is still a wide open area of research. This is not due to a lack of first-rate research by the engineering community, but instead it reflects the complexity of the problem. In order to benefit from the GPUs and overcome these problems, we must go beyond simply "porting" an existing algorithm to the GPUs, which will make us face a much higher bar or look for some novel approaches.

We have implemented our algorithm to the GPU. The transformation part takes the most of the processing time, so we decided to implement transformation part into the GPU and the other parts implemented in CPU domain. Many threads are created and performed action and than the output were written to the CPU. It took more than we expected. For a GA optimization of 100 population and 8 iterations, it took four times more than CPU does. The main issue behind is that the device-CPU communication speed. It takes a lot of time for CPU and GPU to communicate each other. The huge amount of the processing time goes to this communication. The process is shown in Figure 16.

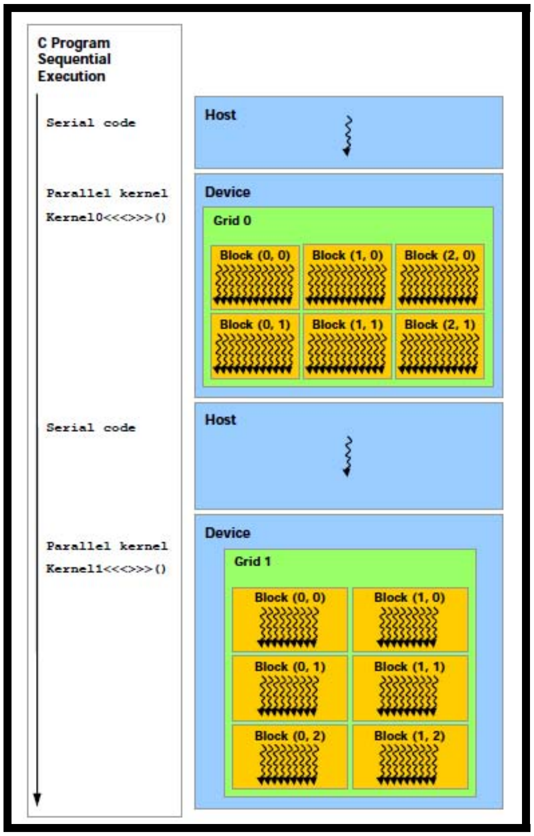


Figure 16: CPU-GPU Communication and Parallel Processing

4. RESULTS AND DISCUSSIONS

The entire algorithm is implemented by using MS-Visual C++ 6.0 and Intel OpenCV. All experiments are performed on a Windows Vista computer mounted with a 2.33 GHz Intel Core 2 CPU and 2GB RAM. The input optical images from Camera 1, 6, and 8 are all 640×480 color images, and the input IR image from Camera-7 is 320×240 dummy color. The discrete value for μ in Eq. (10) is set at 0, $\pi/4$, $\pi/2$, and $3\pi/4$. The quadruplet of the landmarks for rectification is selected from the intersection candidates, manually. The landmark region size is determined by using the Google earth map.

In the first experiment, the landmark region size is a 12m×6m rectangle. This rectangle is located in front of Camera-1, and about 20m away from the building wall. Camera 7 and 6 is about 29m and 51m away from this rectangle, respectively. Figures 5 (a), (b), and (c) show the input image from camera 1, 7, and 6, respectively.

4.1 Experiment-1

The first experiment employed three stationary cameras – two optical cameras (camera 1 and camera 6) and one IR camera (camera 7). The landmark region size is a 12m×6m rectangle. This rectangle is located in front of camera 1, and about 20m away from the building wall. Camera 7 and 6 is about 29m and 51m away from this rectangle, respectively. Figures 17 (a), (b), and (c) show the input image from camera 1, 7, and 6, respectively, and (d), (e), and (f) the rectified images, correspondingly.

The rectified images are overlapped to do image fusion. Because all cameras are top-down cameras, the images from these cameras are largely distorted. The fusion of the rectified image will generate blocking effects at the boundary areas. To reduce these blocking effects, the rectified images are further divided into subimages to perform image fusion. Details are as follows.

For two rectified input image, $\tilde{I}_j$ and $\tilde{I}_k$ (*j*, *k* = camera 1, 7, 8, or 6), the subimages at (*m*, *n*), $\tilde{I}_{j,sub}(m,n) \in \tilde{I}_j$ and $\tilde{I}_{k,sub}(m,n) \in \tilde{I}_k$, are fused in the following way.

(i) If $\sum \tilde{I}_{j,sub}(m,n)$ =0, and $\sum \tilde{I}_{k,sub}(m,n)$ =0, the fused image is also zero;

(ii) If $\sum \tilde{I}_{j,sub}(m,n) \neq 0$, and $\sum \tilde{I}_{k,sub}(m,n) = 0$, $\tilde{I}_{j,sub}(m,n)$ is considered as the fused image;

(iii)If $\sum \tilde{I}_{j,sub}(m,n)$ =0, and $\sum \tilde{I}_{k,sub}(m,n) \neq 0$, $\tilde{I}_{k,sub}(m,n)$ is considered as the fused image;

(iv)If $\sum \tilde{I}_{j,sub}(m,n) \neq 0$, and $\sum \tilde{I}_{k,sub}(m,n) \neq 0$, the fused image is generated according to Eq. (3) or Eq. (7).

Figure 18 (a) shows the fused image for rectified image in Figures 17 (d) and (e) by using DWT based on 4×4 sub-image; (b) the fused image for rectified image in Figures 17 (d) and (e) by using DWT based on 8×8 sub-image; (c) Fused image for rectified image in Figures 17 (d) and (e) by using the linear conjugation based on 2×2 sub-image; (d) the fused image for rectified image in Figures 17 (e) and (f) by using DWT based on 4×4 sub-image; (e) the fused image for rectified image in Figures 17 (e) and (f) by using DWT based on 8×8 sub-image; (f) the fused image for rectified image in Figures 17 (e) and (f) by using the linear conjugation based on 2×2 sub-image, respectively. In all experiments, κ_1 and κ_2 are both set at 0.5.

The moving targets are detected by using the fused image. Figure 19 shows the target detection result at frame 0, 20, 30, and 40, where the purple ellipse shows the convex hull of the target, and green box the circumscribed rectangle. The processing time for image rectification, fusion, and target detection is 1498 ms, 2699 ms for DWT based fusion (718 ms for weighted-add based fusion), and 8456 ms, respectively.

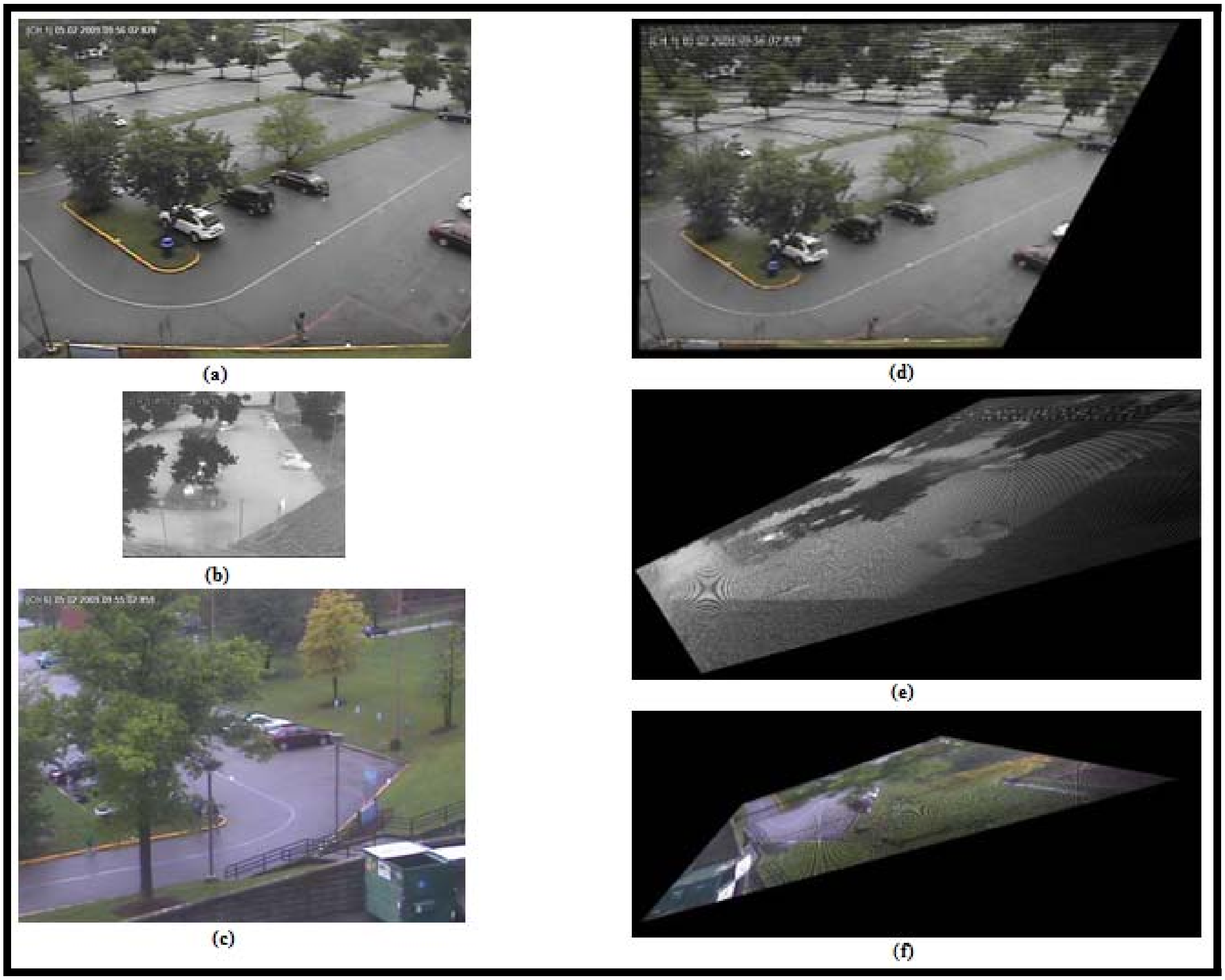


Figure 17: (a), (b), and (c) Input Image from Camera 1, 7, and 6, Respectively; (d), (e), and (f) Rectified Image for Input Image in (a), (b), and (c), Correspondingly

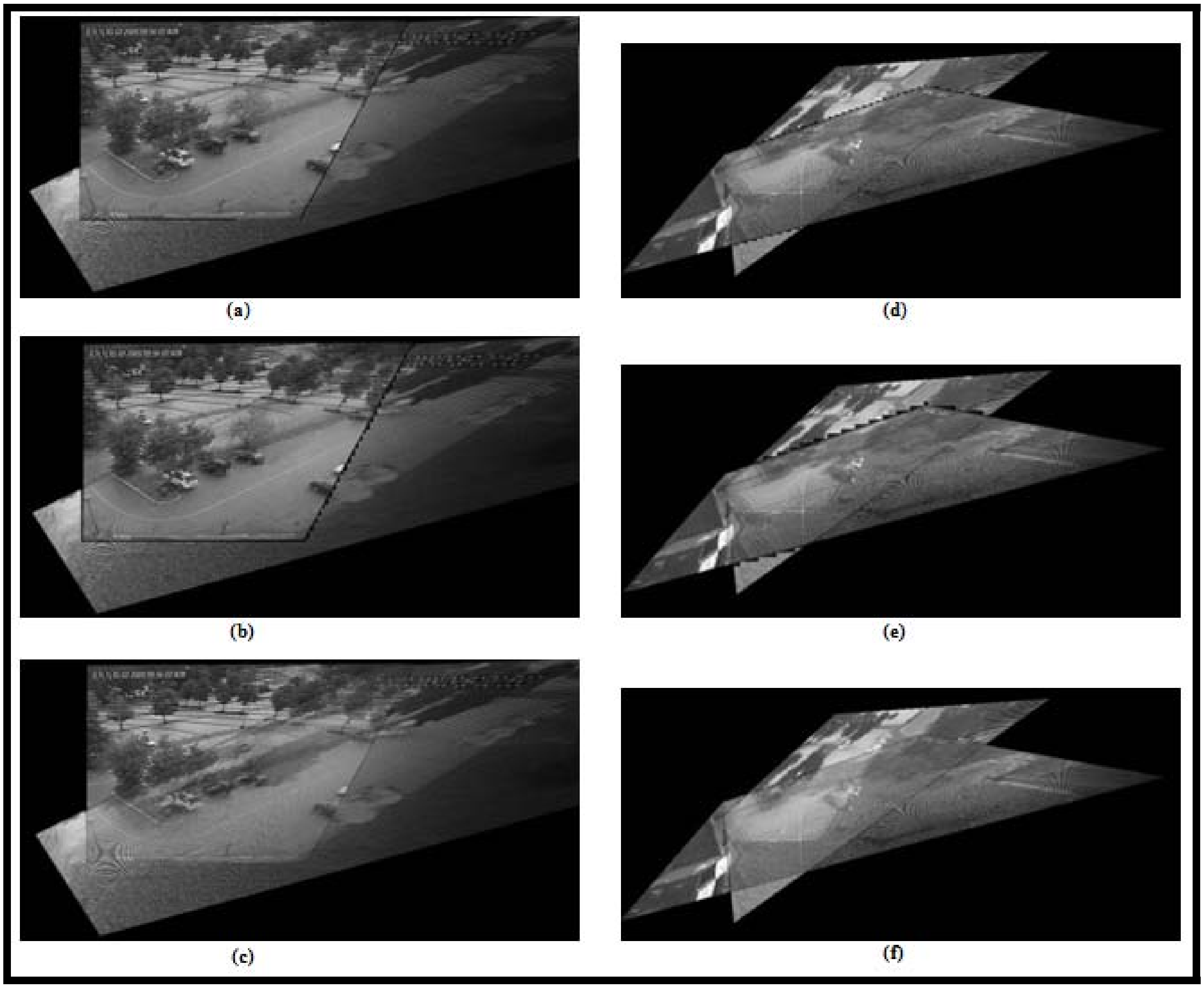


Figure 18: (a) Fused Image for Rectified Image in Fig. 7 (d) and (e) by using DWT based on 4×4 Sub-image; (b) Fused Image for Rectified Image in Fig. 7 (d) and (e) by using DWT based on 8×8 sub-image; (c) Fused Image for Rectified image in Fig. 7 (d) and (e) by Using the Linear Conjugation based on 2×2 Sub-image; (d) Fused Image for Rectified Image in Fig. 7 (e) and (f) by using DWT based on 4×4 Sub-image; (e) Fused Image for Rectified Image in Fig. 7 (e) and (f) by using DWT based on 8×8 Sub-image; (f) Fused Image for Rectified Image in Fig. 7 (e) and (f) by Using the Linear Conjugation based on 2×2 Sub-image

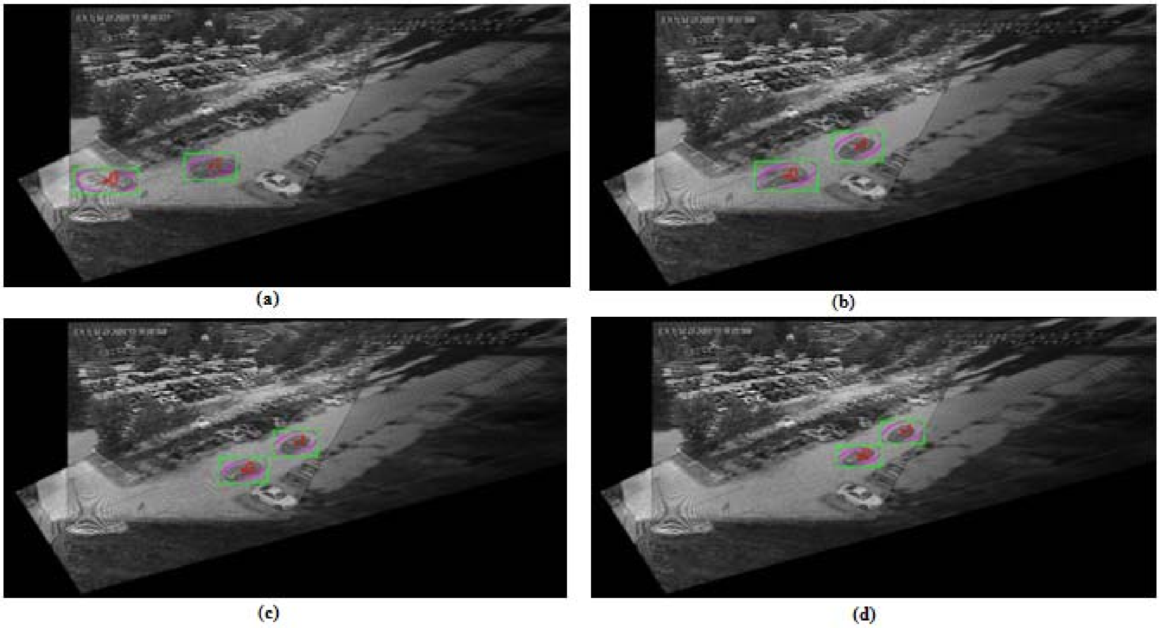


Figure 19: (a), (b), (c), and (d) Targets Detected at Frame 0, 20, 30, and 40, Respectively

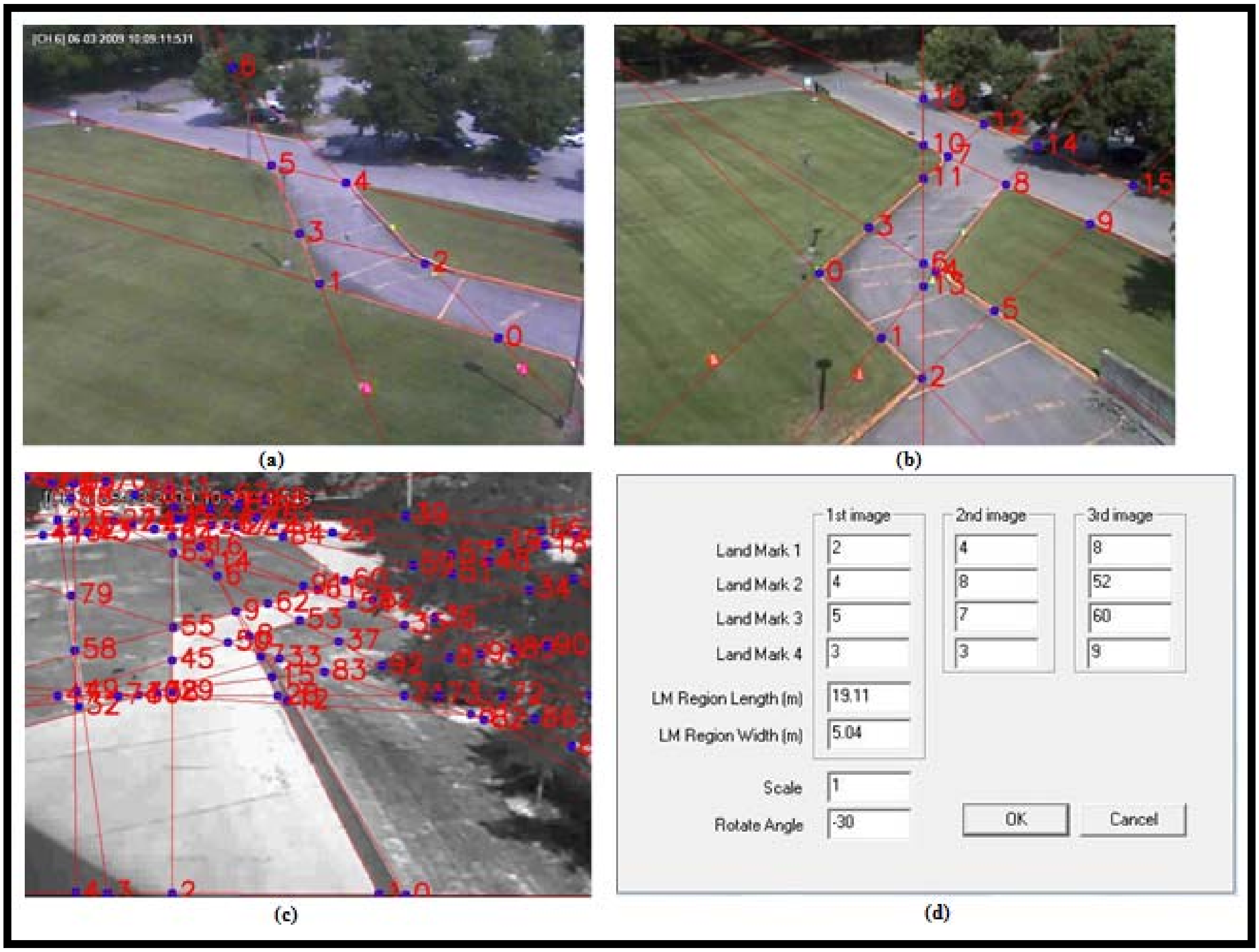
4.2 Experiment-2

The second experiment employed camera 6, 7, and 8, where camera 8 is a hand-held camera. The landmark region size is a 18.96×5.04m rectangle. This rectangle is located in front of camera 8 and 6, and about 20m away from the building wall. Camera 7 is about 39m away from this rectangle, respectively. In this experiment, the processing tasks become more complicated that those in the first experiment. In comparison to the target detection by employing the fused image in previous experiment, this experiment conducts the target detection from the individual video from the independent camera. The processing tasks in this experiment are summarized as follows.

(i) *Processing tasks for video from camera 6*. They contain the projective transformation matrix estimation, image interpolation, image fusion, and moving target detection.

(ii) *Processing tasks for video from camera 7*. They contain the image resampling (because the image size from camera 7 is 320×240 but not 640×480 as those from camera 6 and 8), projective transformation matrix estimation, image interpolation, and image fusion.

(iii)*Processing tasks for video from camera 8*. They contain the landmark tracking, the frame-to-frame transformation matrix estimation, projective transformation matrix estimation, image interpolation, and image fusion. Because the camera 8 is a pan-tilt hand-held camera, it is necessary to track the landmarks so that the consecutive image frame can be registered by using the first image frame as the reference, and then the consecutive image frames can be fused with the images from camera 6 and 8. The registration of the image from camera 8 can be performed by using the affine transformation or projective transformation. Currently because this camera does not contain the rotation, and the moving area is relatively small, the translation transformation is employed.



	1st image	2nd image	3rd image
Land Mark 1	2	4	8
Land Mark 2	4	8	52
Land Mark 3	5	7	60
Land Mark 4	3	3	9
LM Region Length (m)	19.11		
LM Region Width (m)	5.04		

Scale: 1

Rotate Angle: -30

Figure 20: (a), (b), and (c) Initial Image of Three Video sequences from Camera 6, 8, and 7, respectively, where Blue Dots are Landmark Candidates; (d) Quadruplet Selection and Landmark Dimension Selection Interface

Figures 20 (a), (b), and (c) shows the initial image of three video sequence from camera 6, 8, and 7, respectively, where blue dots are land mark candidates; and (d) quadruplet selection and landmark dimension selection interface. In this case, quadruplet {2, 4, 5, 3}, {4, 8, 7, 3}, and {8, 52, 60, 9} from the first frame of camera 6, 8, and 7, respectively, are selected as the landmarks for image rectification. Figure 21 shows the landmark images extracted from the first image of camera 8. These landmarks need to be tracked in the consecutive frames from camera 8. The tracking is based on the template matching technique, and the correlation coefficient is used as the template matching measure.

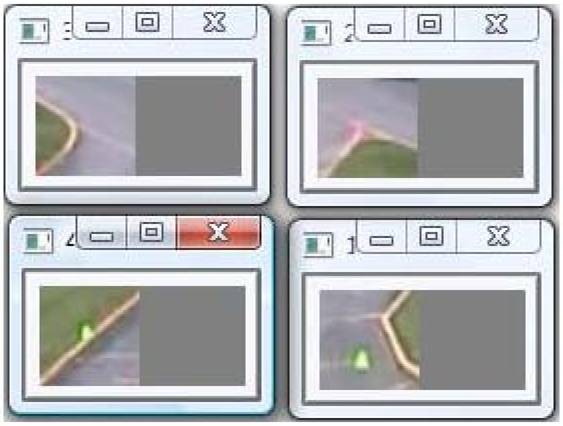


Figure 21: Landmark Images Extracted from the Initial Image Frame From Camera 8, in which the Landmark Candidates 4, 8, 7, and 3 are Selected as the Landmark

Figure 22 shows the image fusion and target detection result at frame 57. The top-left image is the video from camera 6, top-right from camera 7, bottom-left from camera 8, and the bottom-right the fused image which is resized to 640x480. Two moving targets are detected as marked by the purple ellipses, where the green ellipses represent the cluster of the blobs in mask image.

Figure 23 shows the result of the congestion detection at frame 79, which is circled by a white rectangle (see top-left and bottom-right images).

Figure 24 shows the end of the congestion at frame 122, and two moving targets are detected separately (see top-left and bottom-right images).

Figure 25 shows a target disappeared and a new target is detected at frame 202 (see left-top and bottom-right images).

Figure 26 shows the image fusion results for videos from camera 6, 8, and 7; (a) shows the fusion result without interpolation, and (b) with Interpolation.

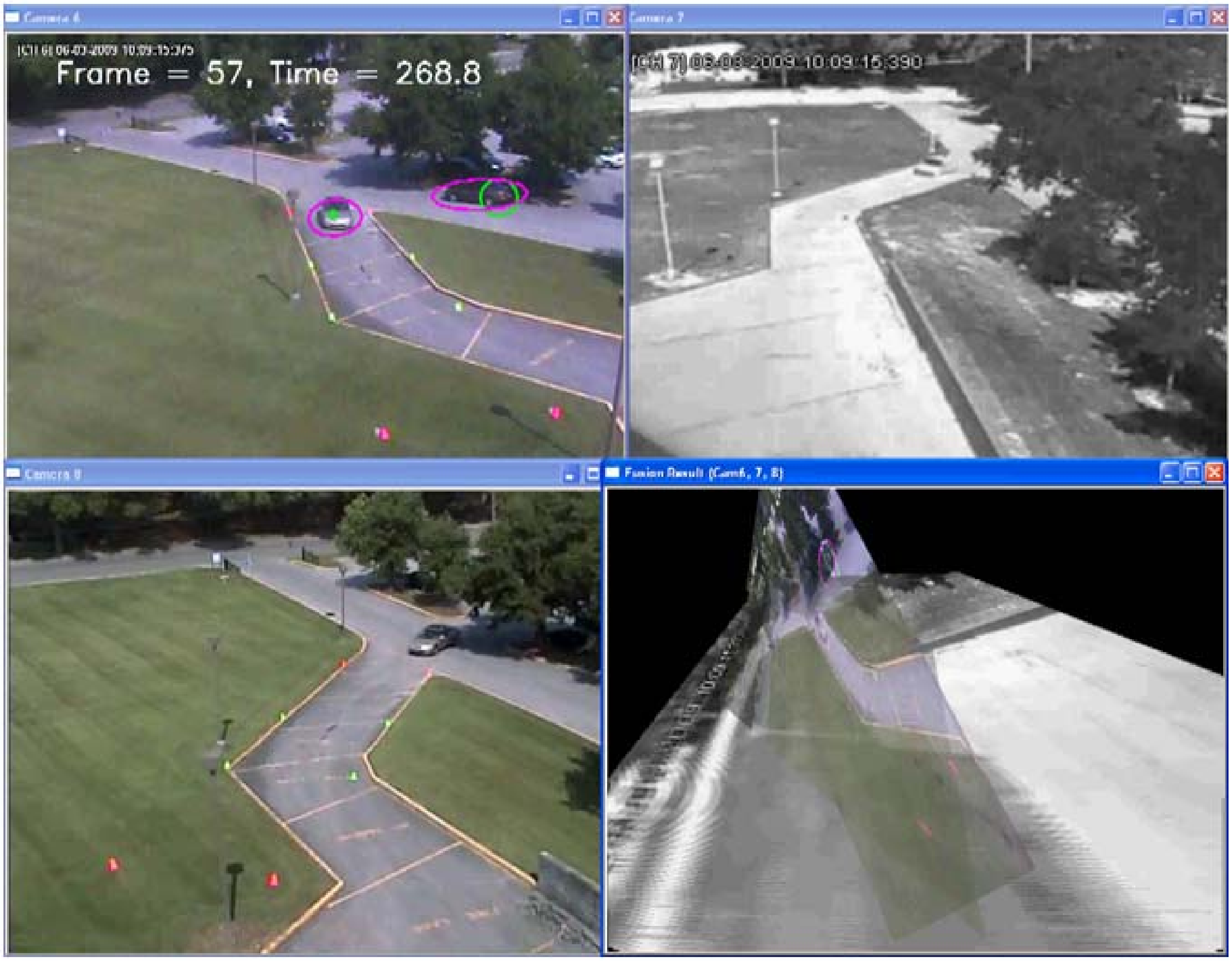

Camera 6
[CH 6] 06-03-2009 10:09:15:375
Frame = 57, Time = 268.8
Camera 7
[CH 7] 06-03-2009 10:09:15:390
Camera 8
Fusion Result (Cam6, 7, 8)

Figure 22: Image Fusion and Target Detection Result at Frame 57. Top-Left Image Is the Video From Camera 6, Top-Right From Camera 7, Bottom-Left From Camera 8, And Bottom-Right The Fused Image which as Resized To 640x480. Two Moving Targets are Detected as Marked by the Purple Ellipses, where the Green Ellipses Represent the Cluster of the Blobs in Mask Image

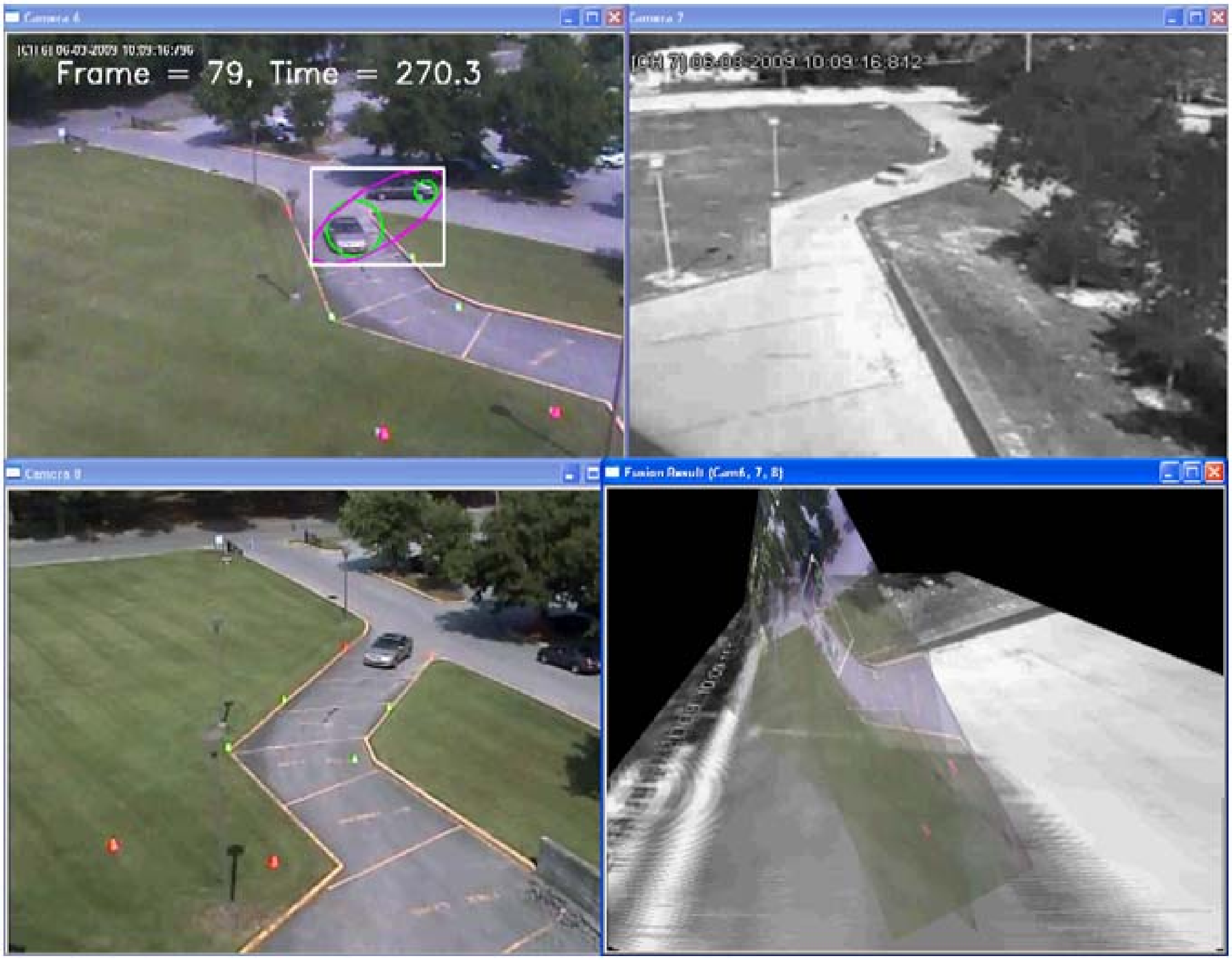
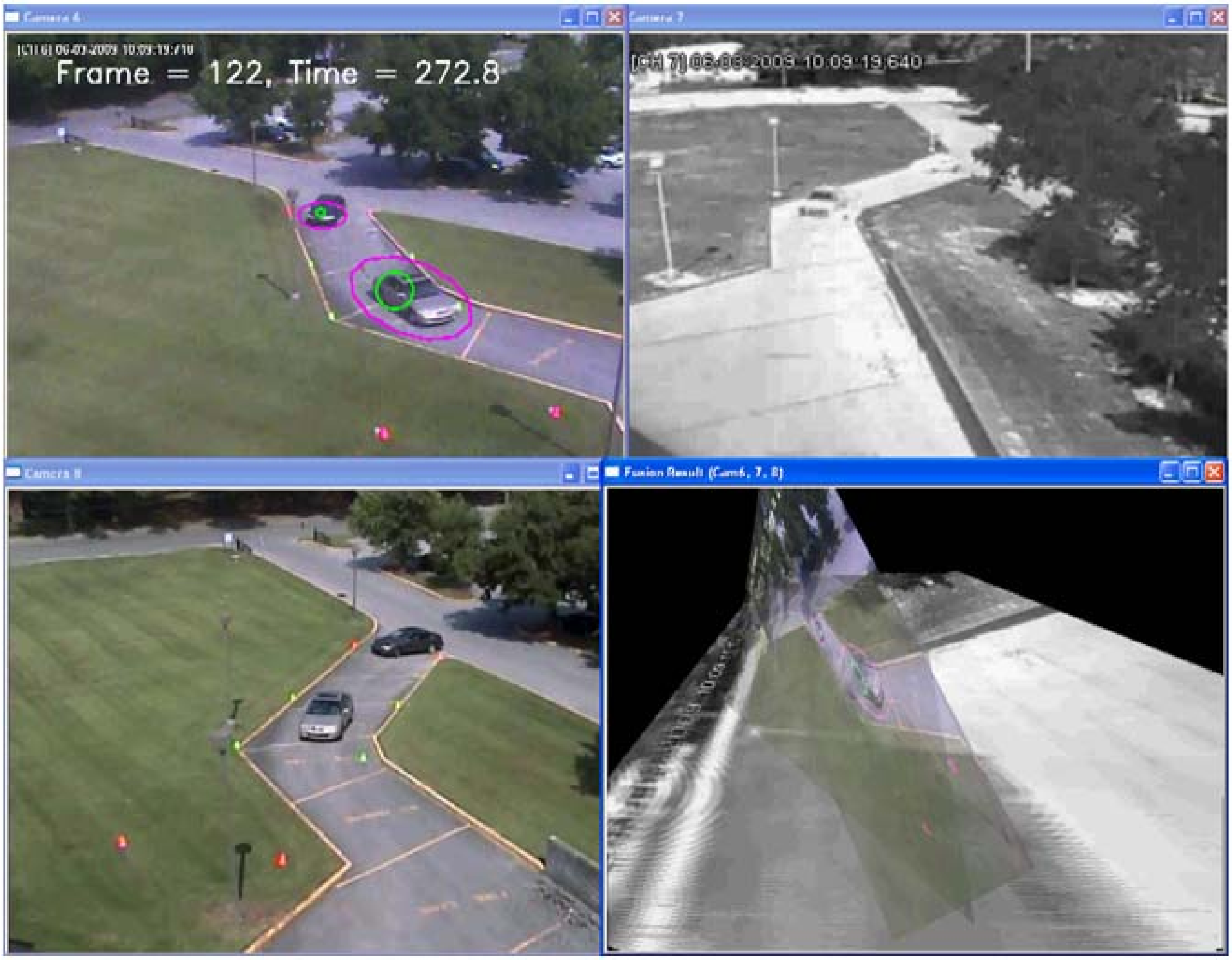

Camera 6
[CH 6] 06-03-2009 10:09:16:796
Frame = 79, Time = 270.3
Camera 7
[CH 7] 06-03-2009 10:09:16:812
Camera 8
Fusion Result (Cam6, 7, 8)

Figure 23: Congestion Detection at frame 79, which is circled by a White Rectangle (Top-Left and Bottom-Right Images)



Camera 6
[CH 6] 06-03-2009 10:09:19:710
Frame = 122, Time = 272.8
Camera 7
[CH 7] 06-03-2009 10:09:19:640
Camera 8
Fusion Result (Cam6, 7, 8)

Figure 24: Congestion Finished at Frame 122, and Two Moving Targets are Detected Separately

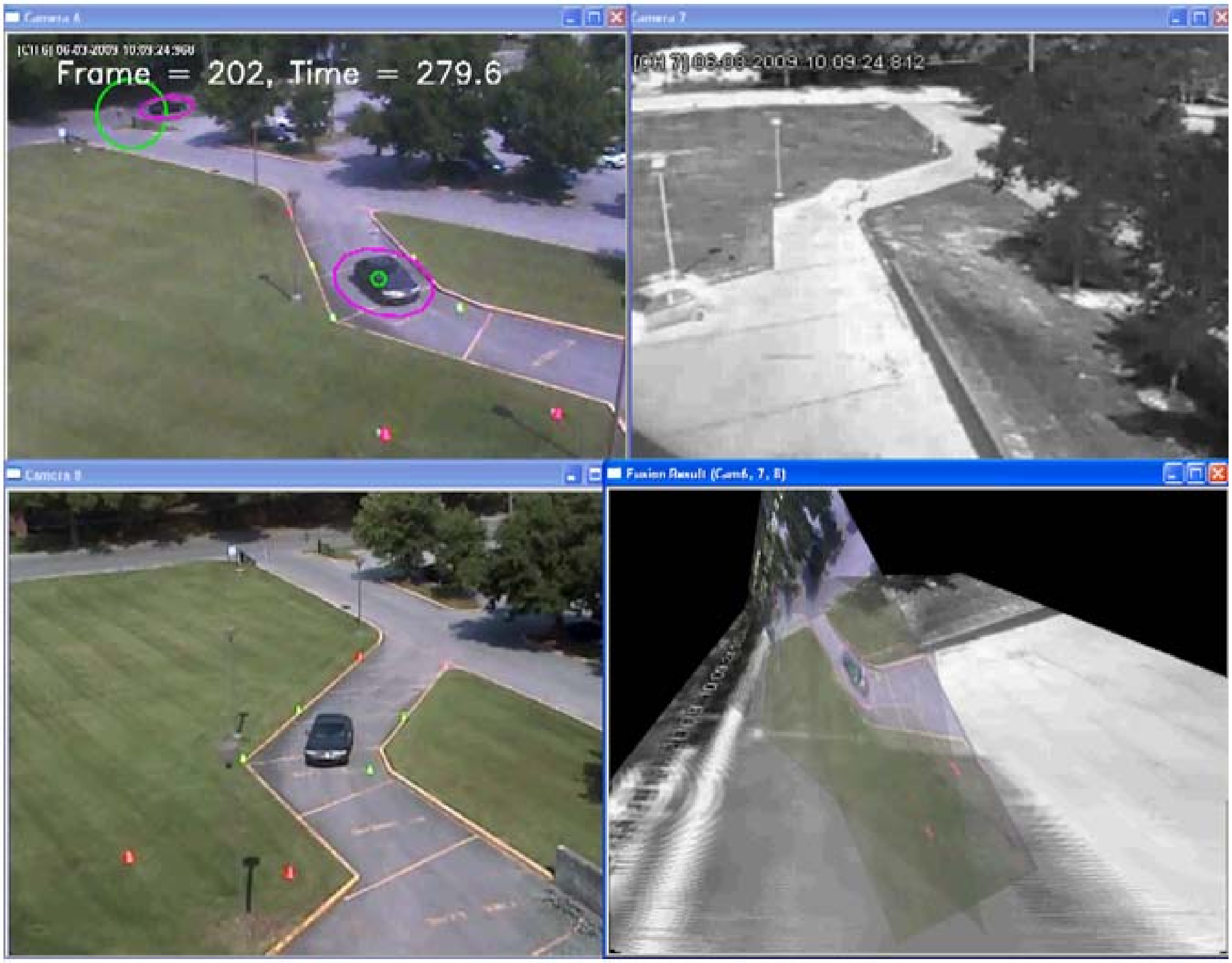

Camera 6
[CH 6] 06-03-2009 10:09:24.960
Frame = 202, Time = 279.6
Camera 7
[CH 7] 06-03-2009 10:09:24.812
Camera 8
Fusion Result (Cam6, 7, 8)

Figure 25: Disappearing of a Target and the Detection of a New Target at Frame 202

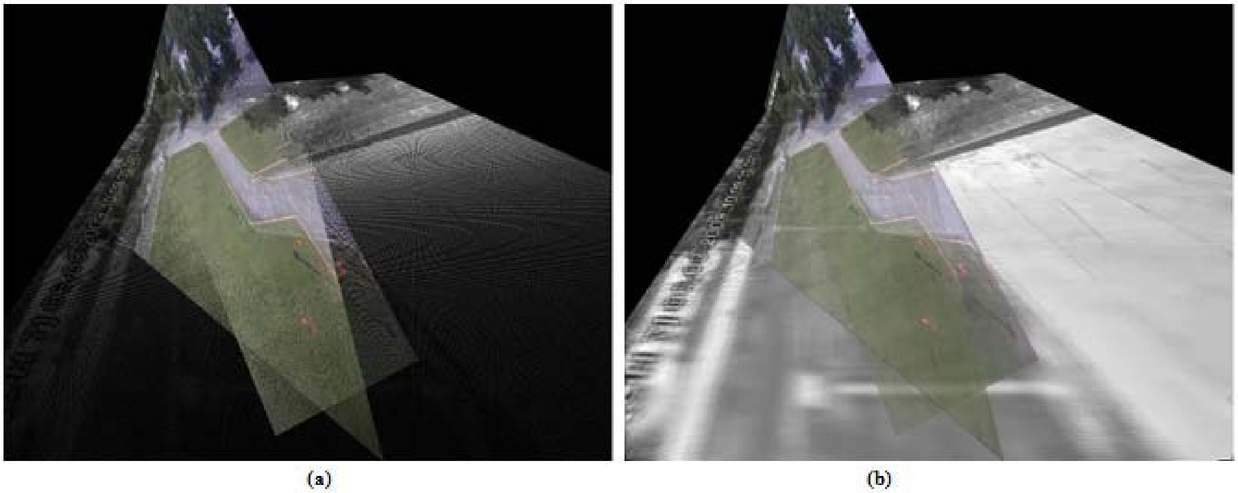


Figure 26: Image Fusion Results for Videos from Camera 6, 8, and 7. (a) No interpolation. (b) With Interpolation

4.3 Experiment-3

The third experiment employed two hand-held cameras which are held by two persons at the roof of 7-story building. The cameras are set 20-meter apart, and are swept left and right to mimic the camera mounted on UAV. In the following, we call them pseudo UAV cameras.

The experiments results are shown in Figure 27 and Figure 28. In this experiment, the lane line and road edge are considered as the land marks. The entire processing contains land mark detection, land mark matching, and image fusion. In both Figure 27 and Figure 28, (a) and (b) shows the input image from pseudo UAV camera 1 and 2, respectively; (c) and (d) shows the detected land marks from the input image in (a) and (b), correspondingly; and (e) shows the fusion result.

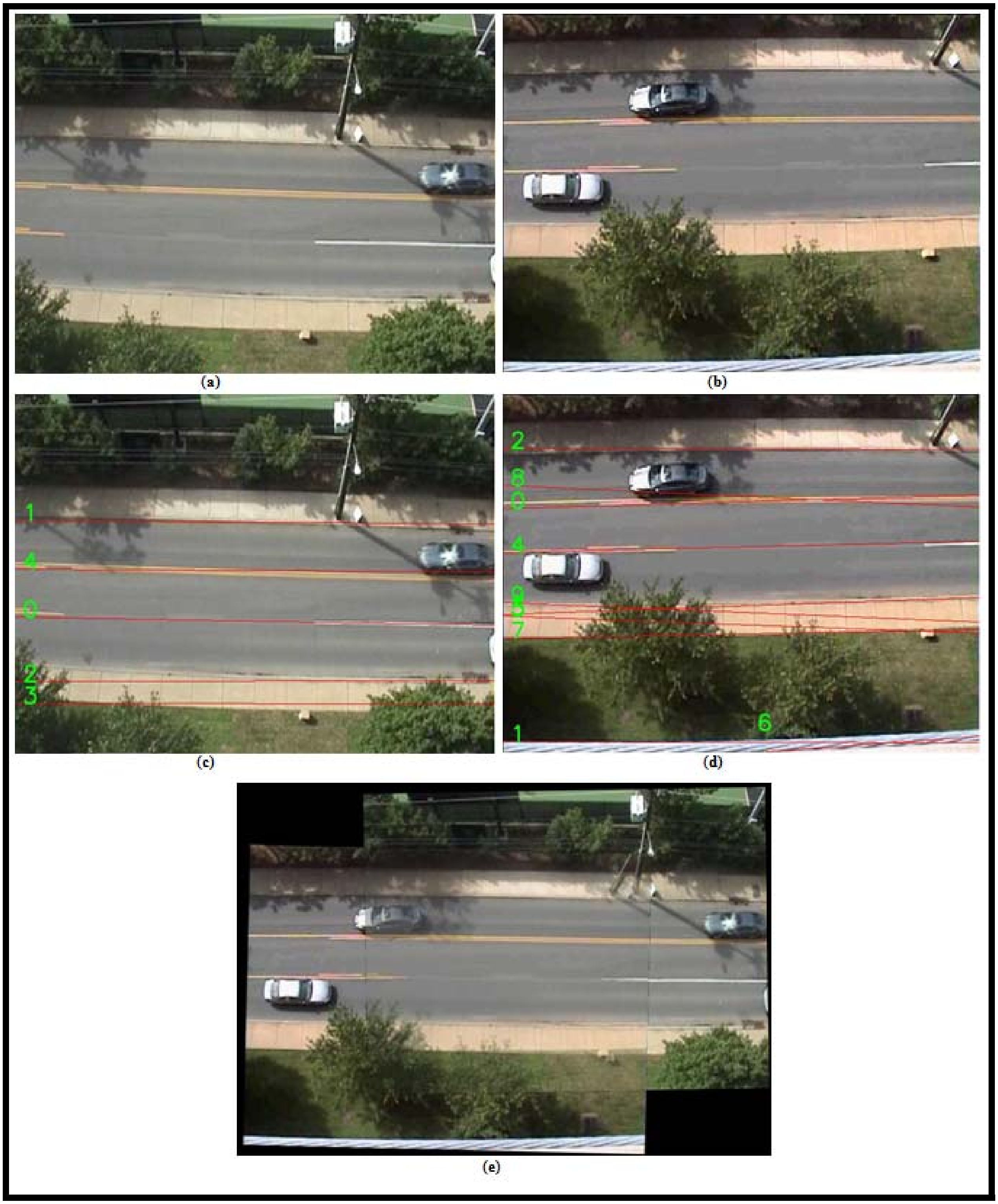


Figure 27: (a) and (b) Input Image from Pseudo UAV Camera 1 and 2, Respectively; (c) and (d) Detected Landmarks from the Input Image in (a) and (b), Correspondingly; (e) Fusion Result

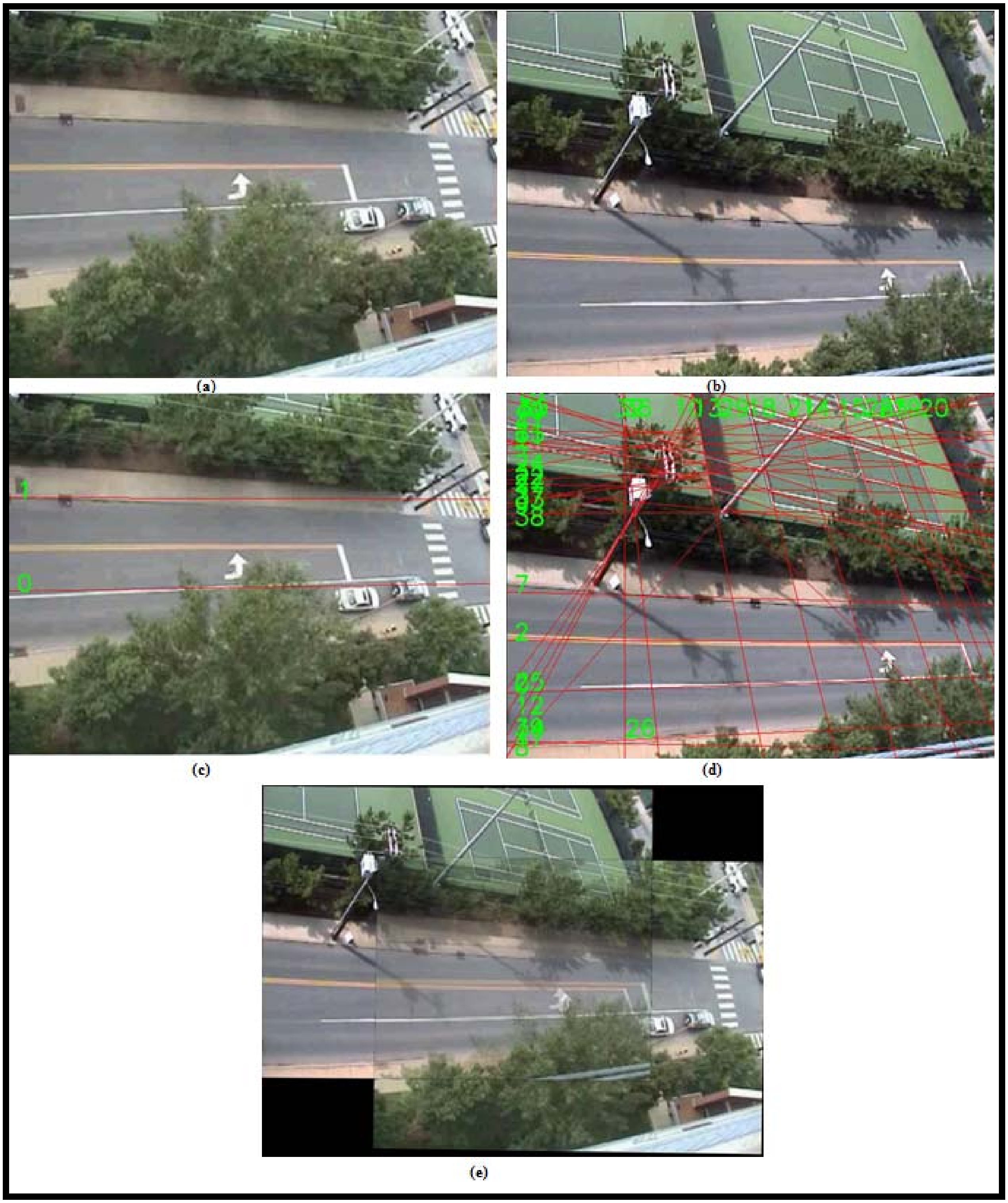


Figure 28: (a) and (b) Input Image from Pseudo UAV Camera 1 and 2, Respectively; (c) and (d) Detected Landmarks from the Input Image in (a) and (b), Correspondingly; (e) Fusion Result for Images in (a) and (b)

5. CONCLUSIONS

This report presented a new method for the fusion of images from heterogeneous top-down cameras. It consists of image rectification, interpolation, and fusion. The image rectification is based on projective transformation, and the image fusion is based on DWT and linear conjugation. It further introduced the integrated Gabor filter for the moving target detection from the fused images. The experimental results verified that the proposed method is valid and effective.

From Figure 5, we can get the following conclusions. Comparing Figure 5 (d) and (e) with (f), for the meaningful overlapped areas (in which the image signal from both input images is not all zeros), DWT-based fusion is smoother than the linear-conjugation based fusion. Same conclusion can be obtained by comparing (g) and (h) with (i). However, the blocking effects of DWT-based fusion are larger than linear-conjugation based method. This is because that smallest subimage size in DWT-based fusion is 2×2, but it can be 1×1, *i.e.*, pixel by pixel fusion for the linear conjugation based method. The processing time for DWT based fusion is approximate four times higher than that of the processing time for the linear conjugation based fusion. This is because that DWT-based method is much more complicated than the linear-conjugation based method.

The second experiment employed the image interpolation to improve the quality of the fused image. Comparing Figures 26 (a) and (b), it is clear that the fused result in (b) is better than that in (a).

5.1 Processing Time

The average processing time for target detection in the first experiment is 8456 ms. This is because that the target detection employs the fused image and the fuse image size is four times of the input image size, in the first experiment. In the second experiment, the processing time is reduced to 3613 ms. This is because the target detector employed the individual input image. The processing in the third experiment contains the landmark detection, landmark matching and image fusion. The average processing time is 9484 ms. The major processing in this experiment is the time for landmark matching. The landmark size is 350×30, and the landmark matching contains ration, scaling and template matching. This processing is repeated in the range of $\varphi \in \{\varphi_{\min}, \varphi_{\max}]$ and $s \in [s_{\min}, s_{\max}]$, where φ is the rotation angle, and s scale. However, this time can be reduced by reducing the size of the landmark.

5.2 Image Fusion Error

From Figure 6 we can see the landmark region areas are well fused, but the perimeter area of each images are not well fused. There are several reasons to cause this error. The first reason is that three cameras are far apart each other, and they are top-down cameras. The second is that these three cameras are different type camera, and their lenses characteristics are different. The third is that the ground is not flat. This type error is also seen in Figure 27 (e) and Figure 28 (e). In Figure 27 (e), the two images from two pseudo UAV cameras are well fused. However, the electric poles in two images are not well fused (see top-right area). This is because the electric pole is not flat to the road surface, and two pseudo UAV cameras are far apart each other. Similar error also exists in Figure 28 (e). This error is unavoidable if the camera lenses are different. But we can conclude that the closer the cameras are, the smaller the error is, and the flatter the terrain is, the smaller the error is. And this error can be greatly reduced by using 3-D camera model.

One of the proposed methods employed the landmark for image fusion. Landmark detection and matching takes long time. However this processing is required at the beginning of the processing. In the consecutive frames, this task becomes the tracking of the landmark. Therefore the algorithm can be greatly speeded up. It will be valid algorithm in the real-time application.

6. REFERENCES

[1] Ehlgen, T. and Pajdla, T., “Monitoring Surrounding Areas of Truck-Trailer Combinations,” *Proc. of the 5th International Conf. on Computer Vision Systems (ICVS 2007),* Bielefeld, Germany, March 21-24, 2007.

[2] Liu, Y. C., Lin, K. Y. and Chen, Y. S., “Bird’s-Eye View Vision System for Vehicle Surrounding Monitoring,” *LNCS*, **4931**, 2008, pp.207-218.

[3] Chen, Y. F, *A Bird-View Surrounding Monitor System for Parking Assistance,* Master Thesis, National Central University, Taiwan, 2008.

[4] Broggi, A., Bertozzi, M. and Fascioli, A., **“***ARGO and the Millemiglia* In Automatic Tour,” *IEEE Intelligent Systems*, **14**, no. 1, 1999, pp.55-63.

[5] Yao, F.H. and Sekmen, A., “Fusion of Airborne Optical And IR Images and Its Application To Target Tracking,” *LNCS 5359*, 2008, pp. 651-660.

[6] Wolberg, G., *Digital Image Warping*, IEEE Computer Society Press, 1994.

[7] Burt, P. J. and Adelson, E., “The Laplacian Pyramid as A Compact Image Code,” *IEEE Trans. Communications*, **31**, no. 4, 1983,pp. 532-540.

[8] Toet, A., “Image Fusion by a Ratio of Low-Pass Pyramid, “*Pattern Recognition Letters,* **9**, no. 4, 1989, pp. 245-253.

[9] Burt, P. J., “A Gradient Pyramid Basis for Pattern-Selective Image Fusion,” *Society for Information Display, Digest of Technical Papers*, 1992, pp. 467-470.

[10] Zhang Z. and Blum, R. S. “A Categorization and Study of Multiscale-Decomposition Based Image Fusion Schemes,” *Proc. of the IEEE,* 1999, pp. 1315-1328.

[11] Li, H., Manjunath, B. S. and Mitra, S. K. “Multisensor Image Fusion Using The Wavelet Transform,” *Graphical Models and Image Processing*, **57**, no. 3, 1995, pp. 235-245.

[12] Wilson, T. A., Rogers, S. K. and Myers, L. R., “Perceptual Based Hyperspectral Image Fusion Using Multiresolution Analysis,” *Optical Engineering*, **34**, no. 11, 1995, pp.3154-3164.

[13] Nakano, H., Yamamoto, S. and Yoshida, Y. *Signal and image processing by Wavelet transform*, Kyoritsu Publisher, Tokyo, 2000.

[14] Daubechies, I. “Ten Lectures on Wavelet,” *SIAM*, 1992.

[15] Yao, F.H., Sekmen, A. and Malkani, M. “A Novel Method for Real-Time Multiple Moving Targets Detection from Moving IR Camera,” “*Proc. of International Conf. on Pattern Recognition*, Tampa, Florida, 2008.

[16] Goshrasby, A., Stockman, G. and Page, C. “A Region-Based Approach to Digital Image Registration with Subpixel Accuracy,” *IEEE Trans. Geosci. Remote Sensing*, **24**, 1986, pp.390-399.

[17] Li, H., Majunath, B.S. and Mitra, S. K., “A Contour-Based Approach to Multisensor Image Registration,” *IEEE Trans. Image Processing*, **4**, no.3, 1995, pp.320-334.

[18] Keller, Y. and Averbuch, A., “Multisensor Image Registration via Implicit Similarity,” *IEEE Trans. Pattern Anal. Machine Intel.*, **28**, no. 5, 2006, pp. 794-801.

[19] Lehmann, T. M., Gonner, C., and K. Spitzer, “Survey: Interpolation Methods In Medical Image Processing,” *IEEE Transactions on Medical Imaging*, **18**, no. 11, 1999, pp.1049-1075.

[20] Davis, J. and Bobick, A., “The Representation and Recognition of Human Movement Using Temporal Templates,” *Proc. Comp. Vis. and Pattern Rec.*, 1997, pp. 928-934.

[21] Bobick, A. and Davis, J., “An Appearance-Based Representation of Action,” *ICPR*, 1996.

[22] Bobick, A. and Davis, J. “Real Time Recognition of Activity Using Temporal Templates,” *IEEE Workshop on Applications of Computer Vision*, Sarasota, 1996

[23] Sheikh et.al “Trajectory Association across Multiple Airborne Cameras,” *Trans. on Pattern Analysis and Machine Intelligence*, **30**, no. 2, 2008, pp. 361–367.

[24] Kang, J., Gajera, K., Cohen, I. and Medioni, G., “Detection and Tracking of Moving Objects from Overlapping EO and IR Sensors,” *Proceedings of the Joint IEEE InternationalWorkshop on Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS’04)*, Washington, DC, July, 2004.

7. LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

CPU	Central Processing Unit
DWT	Discrete Wavelet Transform
EO	Electro-Optical
GA	Genetic Algorithms
GPU	Graphics Processing Unit
IR	Infrared
MEI	Motion Energy Image
MHI	Motion History Image
UAV	Unmanned Aerial Vehicle